Utrecht University

Master Thesis

Systematic Selection Of Threat Modeling Approaches

Author:	*First supervisor:*	*Second supervisor:*
Lennard Marck	Dr. Kate Labunets	Dr. Siamak Farshidi

A thesis submitted in fulfillment of the requirements for the degree of Business Informatics

in the department of Information and Computer Science

14-8-2023
3584 CS Utrecht, The Netherlands

Abstract

Threat modeling is a method for identifying and analyzing security problems early on in the development life cycle. The infancy of the discipline, the absence of a shared scope, and variations in complexity and application all contribute to the challenge for decision-makers to select a threat modeling method and tool. This study proposes a systematic decision-making approach, the core of which lies within a decision model suited to mitigate this challenge. The model facilitates the evaluation of threat modeling methods based on a set of criteria. In its current state 95 requirements and 18 threat modeling methods are mapped. The requirements were extracted and refined by doing an SLR, expert surveys, and interviews. Quality criteria were derived and a preliminary mapping between qualities and requirements was created. The context of the selection in terms of goals, scopes, and preferences was investigated and served as input for creating the final systematic decision-making approach. This approach underwent evaluation through a case study using criteria from the Prat taxonomy. Results indicate that the proposed systematic selection approach has the potential for assisting in making traceable decisions but needs to be further refined and validated. Moreover, the collected data and results of the analyses, and especially the methods, requirements, and quality criteria refined through a multi-phased research protocol, can serve as a foundation for future research.

Acknowledgements

The writing of this thesis as part of my Master Business Informatics has been an adventure. If I would explain this to my friends, I would say: The last 9 months felt like riding one of those imitation bulls on a fair (I do not like real bull riding). During this adventure, I have fallen many times and this has brought a lot of frustration. Even though I felt like giving up sometimes, I did not and I got back up on it over and over again. I feel proud in doing so and I am very delighted to present the result of this journey.

Not only did I learn a lot about threat modeling and decision-making but also a lot about myself. Ironically, I am not a great decision-maker, that's also one of the reasons I liked this project so much. I would have loved to have had a systematic approach for writing this thesis, but as you probably know, it's not supposed to be like that.

Throughout this adventure, my family and friends have been nothing but supportive. I am thankful to my main supervisor Dr. Kate Labunets, for her endless guidance, patience, and trust. She went above and beyond to help me finish this project on time and I really appreciate her for that. I also would like to thank my co-supervisor Dr. Siamak Farshidi, for his help with multi-criteria decision-making. I owe special thanks to Dr. Olga Gadyatksaya, for her knowledge of threat modeling and for helping me organize the cluster of concepts I created.

Finally, I want to thank a.s.r for participating in the case study and providing the environment to work and grow. I would like to thank the team at a.s.r. and especially Joey van den Heuvel for seeing something in me and providing daily guidance not only related to my thesis but also for helping me to shape my path toward the future.

Contents

1 Introduction

The fast growth in IT provides new opportunities. For example, numerous IT innovations and applications have been developed to combat the COVID-19 pandemic [72]. However, the growth in this sector is a double-edged sword, as it also exposes IT systems to an increasing number of attacks [200]. These attacks encompass threats that originate both from external and internal sources within the organisation [159]. The impact of successful attacks can vary significantly, ranging from relatively minor but severe incidents like unauthorized access to an individual's email account, such as that of the janitor, to the catastrophic exposure of sensitive personal data belonging to millions of customers[1]. Take for example Uber, an enormous organisation that in late 2022 has fallen victim to yet another cyberattack. The hacker managed to breach the company's security and gained unauthorized access to critical systems and over one petabyte of sensitive company data, including internal financial information. Especially with a growing dependence on IT infrastructure, the importance of taking action before threats can be exploited is more crucial than ever.

To proactively identify, evaluate, and propose mitigations for these attacks, threat modeling has been developed [203]. Threat modeling is a diversified domain, encompassing a wide range of methods that differentiate, among other aspects, in terms of their application, focus, and perspective [200] but also in input, procedure, outcome, guidance, and tool support [175]. The multifaceted landscape lacks a single unified definition, resulting in numerous distinct interpretations and usages [200]. This poses a challenge in establishing a common basis for method comparison. Additionally, the task of identifying threat modeling methods is made more complex, given the wide variety of terms used throughout the literature to refer to a threat modeling method. E.g., Threat analysis [9], methodology [147], technique [150], framework [37], threat identification [140], and more.

In the scope of this thesis, two existing definitions are combined from a top-down view. On a high level, threat modeling can be defined as *"a process that can be used to analyze potential attacks or*

[1] https://www.theguardian.com/technology/2017/nov/29/uber-security-breach-london-sadiq-khan-users

threats and can also be supported by threat libraries or attack taxonomies" [180]. This definition can be specified using the system evaluation perspective, wherein threat modeling entails representing and analyzing the system architecture, resulting in the identification of potential threats. Given these threats, appropriate mitigation techniques are picked [38].

A significant number of different methods for threat modeling exist. These all have a different technique, such as STRIDE [150], PASTA [177], LINDDUN [37], Attack Trees [155], and plenty more. They utilize a notation and are sometimes supported by a tool [22]. Threat modeling methods and tools are already evaluated and compared in literature [156, 159, 196]. However, due to the inherent diversity of the domain, there is not a single method that fits best for all scenarios [175, 200]. Selecting the "best" method can be accomplished by aligning the organisational context with the application context of the threat modeling method [156]. Factors such as risk, dedicated time, experience with modeling, and stakeholder preferences influence this decision [159]. However, this selection process is still perceived as burdensome and exhausting [156].

The infancy of the discipline, the lack of a common scope, and differences in complexity and application all contribute to the complexity of the method selection challenge [161, 175]. Decision-makers have to deal with a significant number of alternatives, each featuring a distinct collection of components, each possessing a set of diverse characteristics [200]. This selection is conducted distinctively by each stakeholder and the suitability of selection is dependent on the alignment of contextual factors with the threat modeling method. Moreover, The quantity and variety of threat modeling methods significantly increase the selection challenge difficulty. Given these observations, it is particularly relevant to provide adequate guidance. Therefore, this study aims to develop a systematic decision-making approach that utilizes a decision model to capture knowledge about threat modeling methods in a reusable format. To achieve this, multiple research activities are conducted to capture the context of the selection problem. Based on the results of these activities, a systematic approach is designed, which is tested at a large insurance company located in the Netherlands.

1.1 Problem statement

The literature suggests that the threat modeling discipline is still in its infancy [203]. Meaning there are numerous non-validated methods or methods that apply to a specific case study [200]. Several studies try to provide an overview of this domain [156, 159, 175, 200]. Others compare a limited number of methods based on criteria or from the point of view of the supporting tool [22, 161]. In addition to the limited supply of these studies, the comparison or mapping is constructed using a wide range of criteria. The subjects of study share partial overlap, yet a lack of coherence is apparent between the sets of subjects. Moreover, there is a difference in complexity [203] and application of threat modeling, as well as a deficiency in common usage of the definition [200]. All these aspects contribute to the practical challenge for decision-makers to select an appropriate threat modeling method and tool [161, 175].

To address this challenge, several studies offer advice or a recommendation on how their findings translate to this decision-making problem [156, 159, 175]. However, the advice provided by the literature is often oversimplified and fails to account for multiple organisational requirements. For instance, Tuma et al. [175] based their recommendation on one organisational component called *"resource investment"*, which can be "large" or "small". If the resource investment of a practitioner is classified as "small", a set of six general methods and numerous extensions can be considered [175]. Given the guidance provided by the literature, the practitioner is still required to acquire extensive knowledge about various techniques and methods to select a suitable approach [156].

To overcome this challenge and make an informed decision, the practitioner can benefit from a structured approach. As far as known, there is no systematic threat modeling method selection protocol. The selection problem is eligible for the formal specification of relationships between method characteristics and organisational requirements [141]. A formal mapping of the relationships between domain and organisational concepts can be the basis for a systematic decision support method [51]. Nonetheless, this does not exist in the literature, particularly within the domain of threat modeling.

A decision-making process can be defined as human behavior in which a preferred alternative course of action is selected among a set of options, guided by predetermined criteria [188]. Decision-makers in the threat modeling domain are confronted with a significant number of alternatives each encompassing a unique combination of a technique, tool, and notation. Additional complexity arises from the distinct characteristics inherent to each component [200]. To make an informed decision, the decision-maker is challenged with the long-lasting task of evaluating and comparing these alternatives. This task is conducted differently by each stakeholder. The type of system that is modeled and the modeling objective impact this decision, as well as the experience with threat modeling and foreseen time dedication [159]. The substantial quantity and variety of existing methods pose a challenge for practitioners to select an appropriate threat modeling method [175]. Each method has a different focus [86] and therefore is applicable in a certain practical context. Moreover, there is not a single method that is most appropriate for all scenarios. The subsequent subjective selection and evaluation are anticipated to create a difference in judgment among decision-makers. To address these issues Multi-Criteria Decision-Making (MCDM) can be used to create a supporting decision model [52, 55].

MCDM is a domain that encompasses a set of techniques used to evaluate alternative solutions based on the preferences of a decision-maker [40]. The MCDM process contains six activities: 1) identify the object, 2) select criteria, 3) pick alternatives, 4) decide upon weighing method, 5) method of aggregation, and 6) aggregation-based decision-making [108]. This leads to a prioritized list of alternative solutions, arranged from the most suitable to the least suitable alternatives [40]. By creating a decision model that does a mathematical evaluation based on the aggregation function, threat modeling methods can be evaluated over a set of criteria [77]. This evaluation presents feasible alternatives given the preferences of the decision maker, thereby limiting the decision scope and facilitating rapid decision-making.

1.2 Research goal and main research question

There is currently no structured approach toward threat modeling method selection published in the literature. Therefore, this study aims to design and evaluate such an approach, which employs a decision model to encapsulate knowledge about threat modeling in a reusable format. An SLR, expert surveys and expert interviews are conducted to capture the context of the selection problem from both the academic, as well as the practical perspective. Additionally, a suitable framework for systematic decision-making is selected, adapted, and employed. Following the six steps from Majumder [108], a decision model is developed. By integrating the context, particularly the threat modeling goal, scope, and stakeholder preferences, with the decision model, we aim to develop a systematic decision-making approach. This approach is evaluated in a case study at a large insurance company located in the Netherlands.

In accordance with the research aim, the following Main Research Question (MRQ) is proposed:

> **MRQ:** How can organisations systematically select a threat modeling method?

1.3 Contributions and relevance

The following scientific contributions are made:

1. A decision model that assists with the systematic selection of threat modeling methods.
2. An approach for systematical selection of a threat modeling method given organisational requirements, including the supporting tools.
3. Insights into the practical context of threat modeling, including the perceived goals, scopes, and preferences.
4. Findings from a case study in which the application of selection approach is evaluated on, in a large enterprise[2].

Besides having a scientific implication, the research also has practical implications:

1. Traceable decision support for the selection of threat modeling methods.
2. The evaluation and objective selection of threat modeling techniques, tools, and notations from a top-down perspective.

1.4 Document structure

The first chapter has been dedicated to introducing the research problem, goal, and expected contributions. The next chapter (Chapter 2) presents the underlying concepts of the research and related work. The third chapter (Chapter 3) is devoted to explaining all phases of the research method and the corresponding deliverables. Next, the systematic literature review is documented (Chapter 4). Chapter 5 outlines the activities conducted prior to the expert survey and interviews. Subsequently, the results of the pre-interview survey are discussed (Chapter 6). Thereafter, the expert interviews are reported in Chapter 7. In Chapter 8 we explain how the systematic decision-making approach and corresponding decision model are created and afterward, we elaborate on how the approach is tested in a case study (Chapter 9). Lastly, each research activity is discussed, threats to validity are provided and the research is concluded in Chapter 10.

[2] Has more than 250 employees as defined by OECD [169]

2 Background and related work

The following chapter describes the underlying concepts of the research as well as mentions some related work. First, the cyber threat paradigm is introduced and related to other cyber security topics. Thereafter, the fundamentals behind systematic decision-making are discussed. This section especially zooms in on MCDM, as it is one of the main topics of this thesis. Lastly, the related domain of method engineering is briefly mentioned, introducing some core concepts utilized during this study.

2.1 The cyber threat paradigm

As stated in the introduction, threat modeling is a method that allows proactive identification, evaluation, and mitigation selection for threats [203]. The Oxford dictionary [132] defines (cyber)threat as *"the possibility that somebody will try to damage or destroy a computer network, computer system or website by secretly changing the information on it without permission."* However, this is only a fraction of the definition used in the literature. To cause a threat, an attacker can exploit a vulnerability within a system. A vulnerability is a deficiency in the security of an information system that can be exploited.

Vulnerabilities can occur on different abstraction levels. For example, there can be vulnerabilities in the implementation of the system or the procedures in which the system participates [135]. These deficiencies may cause harm when exploited [134]. In the cyber security domain, harm can be viewed as the loss of value [135]. Sometimes this can be measured directly in monetary value. Other times harm can cause an indirect loss of monetary value, for instance, when productivity is lost [118]. According to Pfleeger and Pfleeger (2012), harm can root itself in the loss of availability, integrity, and confidentiality [135]. The loss of these terms regards the unauthorized use (availability), modification (integrity), and observation (confidentiality) of assets protected by the system

[135]. Examples of these assets are databases containing customer information or (virtual) business processes. Maintaining availability, integrity, and confidentiality are basic security principles of any software system. Nevertheless, when viewed from another perspective, these are objects of security threats [135].

The subject that manifests a threat is often referred to as the threat agent [186]. A threat agent can be internal or external to the system. Threats can be classified into three categories: environmental, technological, and human threats [81]. Environmental threats are non-human threats due to natural processes, such as natural disasters. Physical processes on materials cause technological threats. For example, gaining entry into a restricted area, such as a physical data centre, and stealing the hardware, would be a technological threat [81]. When discussing threats in this research, it is mainly referred to as human threats. As the name suggests, these are threats caused by any human and can impact the non-physical system components. The human agents can have a malicious or non-malicious nature [81]. Non-malicious attacks occur due to poor security policies, often caused by poorly trained employees, with the aim not to harm the system [178]. While malicious threats are caused by any human that aims to harm and disrupt an organisation [81].

2.1.1 Risk versus threat

Numerous approaches exist for handling threats. The concept of risk is introduced to understand the surrounding domain and the complexity of threat handling. To define risk and relate it to the cyber threat paradigm, the standardized guide for conducting risk assessment by the National Institute of Standards and Technology (NIST) is used [128]. NIST defines risk as *"a measure of the extent to which a potential circumstance or event threatens an entity"*. To further elaborate, the process of risk assessment contains the identification, estimation and prioritization of risks, based on organisational components, such as assets, after the use of information systems [128]. On one hand, agents exploit threats to cause harm to assets, on the other hand, risk refers to the estimated measure of perceived threat caused by potential exploitation. The measure of risk, although not standardized, is often a function of the level of consequence and probability of the harm occurring [128].

When comparing the risk and threat definitions, it is understood that risk and threat can be separated based on the point of view of the attack problem. Nonetheless, when observing the activities surrounding these concepts, they are often intertwined. Take for example the concepts of risk assessment defined by NIST. The first step regards the identification of threats to an organisation.

This step is also a core component in the threat modeling definition and purpose [38, 203]. To make matters more fuzzy, the term threat modeling, in literature, is used in numerous distinct and incompatible matters [200]. Additionally, risk assessment is a complex domain comprising a multitude of diverse approaches and subdomains.

2.1.2 The threat modeling scope

Defining the scope of the research concept is often done based on the definition. Nonetheless, the cyber security domain has no standardized definition for threat modeling. The lack of common ground in combination with the diverse nature of the domain results in numerous definitions being used in literature [200]. In order to clarify the scope of threat modeling used in this study, this section will explain the position of threat modeling in the cyber security domain, based on literature examples. The position is established by relating the concept to the standardized risk assessment domain. The term risk assessment refers to both risk analysis and evaluation [7]. The CORAS method is a well-accepted method for model-driven risk analysis in literature [104]. It takes eight steps to complete the method, resulting in a list of identified treatments that contribute to lowering the consequence and/or likelihood of an unwanted incident. A threat diagram is created in steps five and six of the method. The threat diagram contains information system assets and their relation to unwanted incidents caused by the threat scenarios they are facing [104]. These threat scenarios are almost identical to attack scenarios (e.g. "Malcode introduced by hacker via email"). Given one of the most widely accepted (and applicable) definitions of threat modeling, *"A process that can be used to analyze potential attacks or threats"* [180]. Assuming that minimal effort is required for acceptance, CORAS as a method would logically classify as a threat modeling technique. This provides minimal proof that threat modeling can be part of risk assessment, which should be considered when performing a systematic scoping of threat modeling techniques. Besides that, this research uses the high-level definition of Uzunov and Fernandez [180] to set boundaries for the threat modeling scope.

2.2 Decision-making

Decisions are made every day, sometimes in a matter of seconds. "Are you going to watch tv?" or "What are you going to eat for dinner?", these decisions occur daily and only have a relatively small impact on the future. However, decision-making becomes more complex when complicating the problem by introducing decisions on a professional level, where choices have permanent consequences. There are various definitions for decision-making applicable to diverse disciplines. By observing commonalities between the definitions, there are at least three concepts to consider [33, 69, 188]. First, a set of *alternative solutions*, define the scope of comparison and contain the result of the decision-making process. Secondly, there are *criteria* that impact the decision and determine the fitness of the alternatives. Lastly, there are *preferences* of a decision maker, that regulates the importance of the criteria.

Threat modeling in the context of this thesis, is a method in the domain of computer security [15], as a part of the software development process. The alternatives are considered to be threat modeling methods. Threat modeling is still at a low maturity level [203]. There are numerous distinct methods, often lacking validation [200]. Additionally, due to the difference in complexity [203] and applications of threat modeling [200], it is expected to have a wide range of criteria. Given these characteristics, it is challenging for decision-makers to select an appropriate threat modeling method [175].

2.2.1 From decision-making to MCDM

When looking at the systematic decision-making discipline, the techniques used for decision-making can be grouped into three categories: (1) Multi-Criteria decision-making, (2) Mathematical Programming (MP) (3) Artificial Intelligence [27]. An overview of these categories and their attributes are shown in Table 2.1. When initially comparing the three categories, MCDM seems to have a preference. This is due to AI requiring the training data to train the algorithm on the decision-making problem. To obtain an acceptable accuracy, it is expected that much data is needed. In the case of the selection of threat modeling, this is missing. When evaluating MP there is doubt if we can accurately represent the reasoning, especially related to qualitative attributes, in a mathematical function. Therefore, MCDM is in the context of threat modeling method selection the most appropriate method.

Upon further investigation, it becomes evident that this categorization does not necessitate exclusivity when classifying a concept. It can be observed that both AI and MP are rather how decisions are made, while in essence, the context still concerns problems in the MCDM space [36, 62, 202]. MCDM is both an approach and a set of techniques to provide an overall ranking of alternative solutions based on the preferences of the stakeholders [40]. It has been applied in various domains [6] and on different abstraction levels. For example, it can be part of a threat modeling method [129] or used in selecting a software product [67]. MCDM has never been applied to support a

Table 2.1: Systematic decision-making categories and their characteristics.

	Input	Prerequisites	Medium	Output	Example types
MCDM	A set of preferences (weights)	A mapping between criteria and alternatives	Aggregation function	Knowledgeable recommendation	Multiattribute utility methods, outranking methods, compromise
MP	A data representation of alternatives	Decide upon a fitting model, based on the assumed relationship	Mathematical model	Best outcome given mathematical model	Linear programming, goal programming, data envelopment analysis
AI	A data representation of a problem	A set of previous problems and their solutions to train the algorithm	Algorithm	Approximate solutions	Neural network, rough set theory, decision tree

selection of threat modeling methods. It is possible that this is caused by the low maturity level of the threat modeling domain [203], as well as the lack of quantitative benchmarks for the supporting threat modeling tools [161].

To further assess the applicability of MCDM in the context of this study, it is necessary to incorporate findings from related domains. Marle and Gidel [110] used MCDM to support office managers in selecting a project risk management method. Furthermore, there are already a set of established instances where MCDM is used in the software development domain. Hanine et al. [67] uses AHP-TOPSIS for selecting ETL (Extract, Transform, and Load) software. Büyüközkan et al. [21] used a fuzzy MCDM for software development strategy selection. Farshidi et al. [52] created an MCDM-based decision support system for blockchain platform selection. These are merely a few instances among the numerous applications of MCDM to a related selection problem. Given this observation, the goal of the research, its abstraction level, the complex decision-making nature, and the number of factors embedded in the problem context, it is proposed to use the MCDM methodology as a base for systematically solving the threat modeling method selection problem.

2.2.2 Multi-criteria decision-making

MCDM helps select the best alternative given an elaborate set of criteria. The most suitable alternative is selected based on analyzing the application of criteria to alternatives, determining criteria weights for a specific instance and finally ranking the alternatives based on a MCDM technique or formula [6]. Generalizing the MCDM process exposes six activities: (1) object identification, (2) criteria selection, (3) picking alternatives, (4) deciding upon a weighing method, (5) method of aggregation, (6) aggregation-based decision-making [108]. Even though all methods can be generalized to this six-step approach, the MCDM methodology maintains various MCDM methods. A few widely applied examples include Analytical Hierarchical Processing (AHP) [176], Analytical Network Processing (ANP) [181] and TOPSIS [14].

The tools and techniques that apply these methods are decision models that use an aggregation function on each alternative. In the context of MCDM, alternatives are represented as a set of criteria values, that can be applied to any decision-making context. The core attribute of an MCDM decision model is that it provides decision support, given a certain context or set of preferences [173]. To extract this context from the problem domain, the decision maker has to play an active role [41]. An interactive modeling procedure is required to translate the decision maker's domain preferences into the decision model's weight component. An aggregation function calculates a score using both the alternatives and the preferences as input. Next, prioritization based on a score is performed to extract a ranking that can support decision-making [52]. It is worth mentioning that support is a key concept in MCDM, meaning that MCDM can be seen as an informative process rather than an actual decision-making activity. Given the outcome of an MCDM technique, a decision maker can make an informed decision.

2.2.3 Selecting a systematic decision-making framework

Selecting a decision-making technique is often done based on convenience correlating to the knowledge possessed by the researcher or highly influenced by the availability of tool support [77, 94]. To justify the choice of method in this research, the generalised framework for multi-criteria method selection [192] is used. This framework provides support in selecting the appropriate MCDM method given a decision situation.

The framework decides based on four main aspects: weights, performance scale, uncertainty, and decision nature. The lack of quantitative benchmarks in threat modeling [161] motivate using weighing based on preference. Weights can be relative, quantitative, or qualitative [192]. Using relative weights eliminates the homogeneity problem. However, relative weight assessment is a very complex issue [96]. Approaches based on relative weights, such as AHP, have issues with scalability [133]. These issues are rooted in pairwise comparison, which typically is considered as the weight assessment method for extracting relative preferences [51]. Pairwise comparison is very time-consuming [133]. It requires exponentially increasing comparisons when criteria increase [144]. There are solutions to reduce this complexity. However, current solutions can deal with a relatively small number of items [95]. The low maturity of the threat modeling domain [203] is assumed to go together with an increasing number of methods and rapidly changing criteria. Therefore, it is chosen to disregard approaches dealing with relative values.

Weights represent the preference of method and feature requirements in the threat modeling domain. The assessment of preferences is utilized to prioritize the requirements. Preference is subjective [164] and therefore hard to quantify accurately, especially when preferences from multiple stakeholders need to be combined [61]. In software production and method engineering, requirements can be prioritized using qualitative methods [17, 174]. Given these observations, weights in the MCDM for threat modeling are preferred to be qualitative. When applying these criteria to the generalized framework, six out of 56 methods hold. This already provides an indication that there might be a lack of applicable MCDM methods. When furthermore investigating the suitability of the subset of methods, it was found that every MCDM has its strengths and weaknesses. Table 2.2 shows an overview of the MCDM methods, their characteristics, strengths, and weaknesses. Although all of the strengths and weaknesses are considered, the following paragraph will provide a summary of the critical factors which lead to the selection of an MCDM framework. Most of the methods use outranking to compare the alternatives against each other. Meaning that at the end of the aggregation phase, items are compared against each other to create a ranking. There are multiple methods for this. For example, when a cardinal score is calculated for each alternative, score-based outranking can create an order of items based on the implicit sequence. Another commonly used method is to use pairwise comparison to compare alternatives against each other. This allows for comparison with a more flexible input, allowing qualitative or mixed criteria data as input. However, as discussed at the start of this section, a pairwise comparison is not very suitable for a complex decision-making problem. ARGUS [35] and MELCHIOR[106] use this approach to finalize their ranking, thus having scalability issues when the number of alternatives increases.

Similar complexity issues occur while using REGIME [73]. REGIME uses comparison based on the probability of dominance for each combination of alternatives, which besides scalability issues, also increases the complexity of interpretation. The most flexible method is named QUALIFLEX. This method can use a wide range of diverse data types as input. However, this pairs with an increasing complexity [139]. Although this can be overseen, the main issue is a consequence of the ranking method. QUALIFLEX uses score-based outranking, based on the score for every potential order of

Table 2.2: MCDM methods, their characteristics, strengths, and weaknesses.

	Weights	**Criteria values**	**Aggregation function**	**Scalability**	**Comparison**	**Strenghts**	**Weaknesses**
Lexicographic method	Ordinal	Quantitative & qualitative	Relative Criteria ranking	High	Single criteria	1. Simple reasoning method 2. Low time consumption	1. The use of both qualitative and quantitative criteria gets complex really fast [139] 2. Decision can be made based on a single criterion [53] which may not be accurate 3. Relative criteria ranking
ORESTE	Ordinal	Relative ranking	Sum of distance function	Medium	Outranking (score based)	1. Aggregation function is of linear complexity	1. Criteria can be incomparable 2. Choice of distance measure requires a vast knowledge of decision-making method and scope 3. Requires a relative ranking of criteria
MELCHIOR	Ordinal	Any datatype with indifference and preference thresholds	Concordance and lack of discordance	Low	Outranking (pairwise comparison between alternatives)	1. Different types of preference relationships between alternatives provide a multi-dimensional view 2. Deals with any type of criteria data	1. Outranking based on pairwise comparison between alternatives 2. The use of pseudo-criteria require a translation step based on domain knowledge
REGIME	Ordinal	Qualitative	Based on rank order frequency matrix	Low	Outranking (pairwise comparison)	1. Differentiates between negative and positive criteria values	1. Comparison based on probability of dominance for each pair of alternatives 2. Determines suitability based on probability, which makes interpretation extra complex
ARGUS	Ordinal	Preferences (ordinal)	Concordance and lack of discordance (preference graph)	Low	Outranking (pairwise comparison between alternatives)	1. Able to represent all types of criteria in terms of preference	1. Outranking based on pairwise comparison between alternatives 2. Approach is solely based on preference 3. Not reuseable, both weights and criteria values need to be determined in a new case
QUALIFLEX	Quantitative & qualitative	Quantitative & qualitative	Multi-level concordance and lack of discordance	Low	Outranking (score per order)	1. Deals with qualitative and quantitative weights 2. Deals with qualitative and quantitative criteria 3. Concordance and discordance is calculated on three abstraction levels	1. A score is calculated for every potential order which goes bad very fast when you have many alternatives 2. The use of both qualitative and quantitative criteria gets complex really fast [139]
Farshidi	Nominal	Binary & Numeric	Weighted sum method	High	Outranking (score based)	1. Simple computation [112, 121] 2. Integrates preferences from multiple stakeholders 3. Quality criteria are indirectly included in the function 4. The decision model is reusable in similar case studies	1. Only a basic estimate of one's preference function [112, 121] 2. Has a specific application in software production

alternatives. This results in a $O(n^2)$ complexity where n represents the number of alternatives.

Scalability-wise, the lexicographic method [53] is the opposite of these methods. It requires a relatively simple reasoning process where the best alternative is chosen given the top-ranked criterion. Ties are solved by evaluating the next-in-line criterion until one alternative is left [53]. Besides having the domain knowledge to accurately rank every criterion, having a large list of criteria and very heterogeneous alternatives (such as in threat modeling) might result in an early stop. In an early stop, a large portion of criteria is disregarded resulting in an inaccurate representation of the decision-making problem.

When looking for a scalability middle ground, we found ORESTE. ORESTE uses an aggregation function with a linear complexity [145]. A drawback of the method is that it requires a relative ranking of criteria. Here, it can be questioned if the importance of features, in the threat modeling domain, can be generally ranked based on relative assessment. Also, score calculation is based on the general appliance of a distance measure. The selection requires a vast knowledge of the decision-making method and scope. Also, some criteria may be incomparable with each other [145], which makes the method more complex.

In addition to the subset of methods not being optimal in the research context, the general application is also far from being related to the ICT domain. When specifically looking into this domain, the framework more applicable in the given context is explained in the recent doctoral dissertation of Siamak Farshidi [51]. The doctoral thesis contains a framework for creating multi-criteria decision-making models in software production. Example instances are the selection of database management systems, blockchain platforms, and cloud service providers. These instances are typically software systems used continuously. This partially aligns with the threat modeling scope; however, some adaptation would be necessary. The utilized score calculation based on weighted-sum allows for a simple computation [96]. In contradiction to other weighted-sum methods, Fasrshidi [51] incorporated a method of extracting group preferences. Also, a different approach is taken regarding quality criteria. These are generally applicable and have a relationship with features. Therefore, it is suggested to use the framework of Farshidi [51] as a basis for the decision model.

2.3 Method engineering

Method engineering is the scientific domain of formal study of methods, tools, and adaptation of methods [17]. It includes designing and constructing methods, techniques and tools for developing information systems. Some domain contributions are method assembly [18], combining multiple methods, and situational method engineering [68], in which a method is created for a specific implementation context. At first glance, this looks very applicable to the problem addressed in this thesis. However, due to the low maturity of the threat modeling domain, we decided to limit the scope to selecting established methods. Additionally, a limitation is that most of the method engineering contributions require using a method base [19, 68]. As far as known, there are no systematic methods for threat modeling method selection, implying that a significant portion of the suggested processes can not be used in this research. However, a big contribution of the method engineering domain is the standardization of definitions and relations between concepts that are commonly used in this proposal.

> **Method** - A method is an approach to execute a systems development project, structured systematically by means of activities, that correspond with development products [17].

In the context of this research, threat modeling is a part of a software development project. The outcome of threat modeling, in the form of a report of the threats and proposed mitigations, is considered to be a development product.

> **Activity** - An activity consists of tasks or basic activities. These tasks can have conditional execution dependence and a required order [17].

For example, some of the threat modeling activities require the creation of a system architecture to start identifying threats [38]. The activity of creating such architecture, can be considered a preliminary task. The most basic instance of an activity, or standard activity, is a predefined task resulting in a product utilizing some technique and can be supported by some tool [193].

> **Technique** - "A technique is a procedure, possibly with a prescribed notation, to perform a development activity." [17]

When system entities are modeled, they can be visualized in a diagram, table, or graph. Otherwise, they can be formalized; in that case, a text-based representation is used [161, 200]. STRIDE, PASTA, LINDDUN, and numerous additional threat modeling techniques are often bound to a specific notation.

> **Tool** - "A tool is a possibly automated means to support a part of the development process." [17]

In the context of threat modeling, Microsoft Threat Modeling Tool (TMT) is the most mature tool, and supports STRIDE [161]. However, there exist numerous alternative options supporting different techniques.

3 Research method

This research aims to create a systematic approach toward threat modeling method selection. Chapter 2 provided the background on the domain of application, as well as an overview of frameworks for systematic decision-making. Drawing upon this knowledge, this chapter presents the research method.

The research method is segmented into five phases. Starting off with a Systematic Literature Review (SLR), subsequenct stages involve expert surveys followed by subsequent expert interviews. Analyzing the data collected throughout these phases, the problem context is established and a systematic selection approach is designed using the design science framework by Wieringa [195]. This approach is centered around the decision model. Finally, the approach is piloted and validated in a case study. The outcome of these phases will answer the main research question: *How can organisations systematically select a threat modeling method?*. Figure 3.1 provides an overview of the phases and main activities. To answer this question, it is decomposed into several sub-questions:

SRQ1: How can existing threat modeling approaches be compared?

The first question (SRQ1) is dedicated to understanding the fundamental aspects of threat modeling method selection. The outcome of this question should provide an overview of the previously applied mechanisms for their comparison. Additionally, an answer to the following inquiry should be acquired: *Which threat modeling methods, techniques, and tools are documented in existing literature?* The outcome forms the foundation for the alternatives within the decision model. model.

SRQ2: What are the criteria for selecting threat modeling approaches?

In addition to clarifying the scope of the decision in terms of alternatives, a set of comparison criteria is extracted. To answer this sub-question (SRQ2), a list of comparison criteria is derived from the literature and subsequently refined and extended through surveys and interviews with experts from

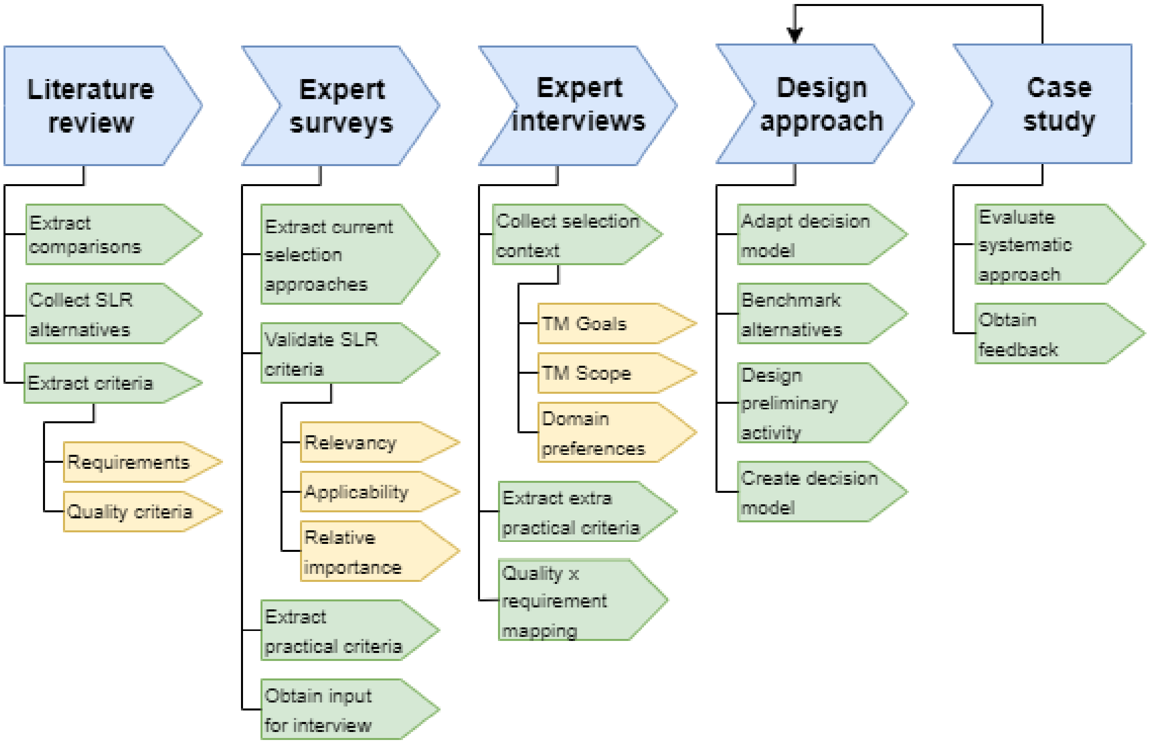

FIGURE 3.1: Overview of the phases and main activities in the research method.

a multi-organisational perspective. This provides an answer to the following inquiries: *"What threat modeling requirements should be considered?"* and *"What are the quality criteria that are used to evaluate threat modeling methods?"* To clarify, the decision-making criteria consist of quality criteria and requirements. The difference between them is further elaborated on in Section 4.7. This question will yield a refined set of decision-making criteria along with their respective importance.

SRQ3: What type of systematic approach can be used for threat modeling method selection?

Once the contents of the decision model have been established, the model is employed. The decision model is adapted from the MCDM framework selected from the literature (See Section 2.2.3). Furthermore, the decision model is populated with a selection of benchmarked alternatives. This provides an answer to: *"What threat modeling methods fulfill which requirements?"* The decision model for threat modeling method selection is the core of the systematic decision-making approach.

Additionally, the practical context of threat modeling method selection is explored in the interviews. This addresses the following inquiry: *What are the domain preferences, threat modeling goals, and threat modeling scopes from practice?* These results in combination with the answer to: "*What current selection approaches are enrolled in practice?* will inform the design of supporting activities and experimental adaptations, to create the final approach and answer the corresponding question (SRQ3).

SRQ4: How well is the proposed decision-making method perceived by the stakeholders?

The last sub-question (SRQ4) aims to evaluate the use of the established approach in practice. In response to the following inquiry: *"How can we evaluate the systematic approach?"* A set of evaluation criteria is defined, which can be used to assess the systematic selection method. An evaluation method is introduced, encompassing the enrollment of the approach within a pilot case. The participants will answer a post-task survey in which they are asked to rate statements related to effectiveness, validity, usefulness, ease of use, completeness, and functionality. The results provide insight into the performance of the approach and its applicability in practice. Moreover, feedback should be obtained that can guide further refinements.

An overview of the research questions and corresponding research methods can be found in Table 3.1.

Table 3.1: Overview of the sub-research question and research method correspondence.

RQ	**SLR**	**Survey**	**Interview**	**Design science**	**Case study**
SRQ1	X				
SRQ2	X	X	X		
SRQ3	X	X	X	X	
SRQ4				X	X
MRQ	X	X	X	X	X

3.1 Literature review protocol

To create a scientific foundation for the decision-making method, a systematic analysis of the literature is conducted. In addition to offering an overview of the existing threat modeling method comparison research, the objective of this analysis is to extract an initial collection of threat modeling tools, methods, and techniques, as well as decision-making criteria for selecting threat modeling methods.

The review adheres to the guidelines by Kitchenham [91]. By utilizing these guidelines, a framework is established, enhancing the comprehensiveness of the literature search. Eleven activities are conducted and divided into three phases: planning, conducting, and reporting. Figure 3.2 Shows an overview of these activities.

3.1.1 Planning the review

Using the previously defined research questions and goals, the objective of the systematic literature review is set. Throughout the planning phase, the corresponding research questions have been reformulated multiple times. An initial exploratory investigation into threat modeling was done using Google Scholar. It was hypothesized that a substantial amount of threat modeling literature regards an adjustment of existing or the creation of new threat modeling methods for niche applications. This indicates the need for a literature review plotting the existing threat modeling landscape [91]. However, this has been done thrice in the past five years [159, 175, 200], two of which were done systematically. Analyzing and comparing these papers indicates that there is no common structured protocol for practitioners to compare or select threat modeling methods.

After the exploratory search, the protocol is developed by defining the data sources, search terms, acceptance criteria, and data extraction method. These are subsequently reviewed and validated in the fourth activity. This entails evaluating the quality of the search query results against a set of papers identified as having "high relevance" during the initial exploratory investigation. After multiple iterations, the contents were deemed sufficient. The final protocol artifacts can be found in Section 4.2.

3.1.2 Conducting the review

The second phase starts by executing a database search, followed by the identification of relevant papers through the application of inclusion and exclusion criteria. A spreadsheet is utilized to collect relevant attributes such as the title, URL, number of citations, year of publication, publication type, venue, keywords, etc. Using the identified papers as a starting point, backward snowballing is conducted to expand the search range. Next, the primary studies are selected by evaluating their relevance to the research questions and objectives defined in the planning phase. This is achieved

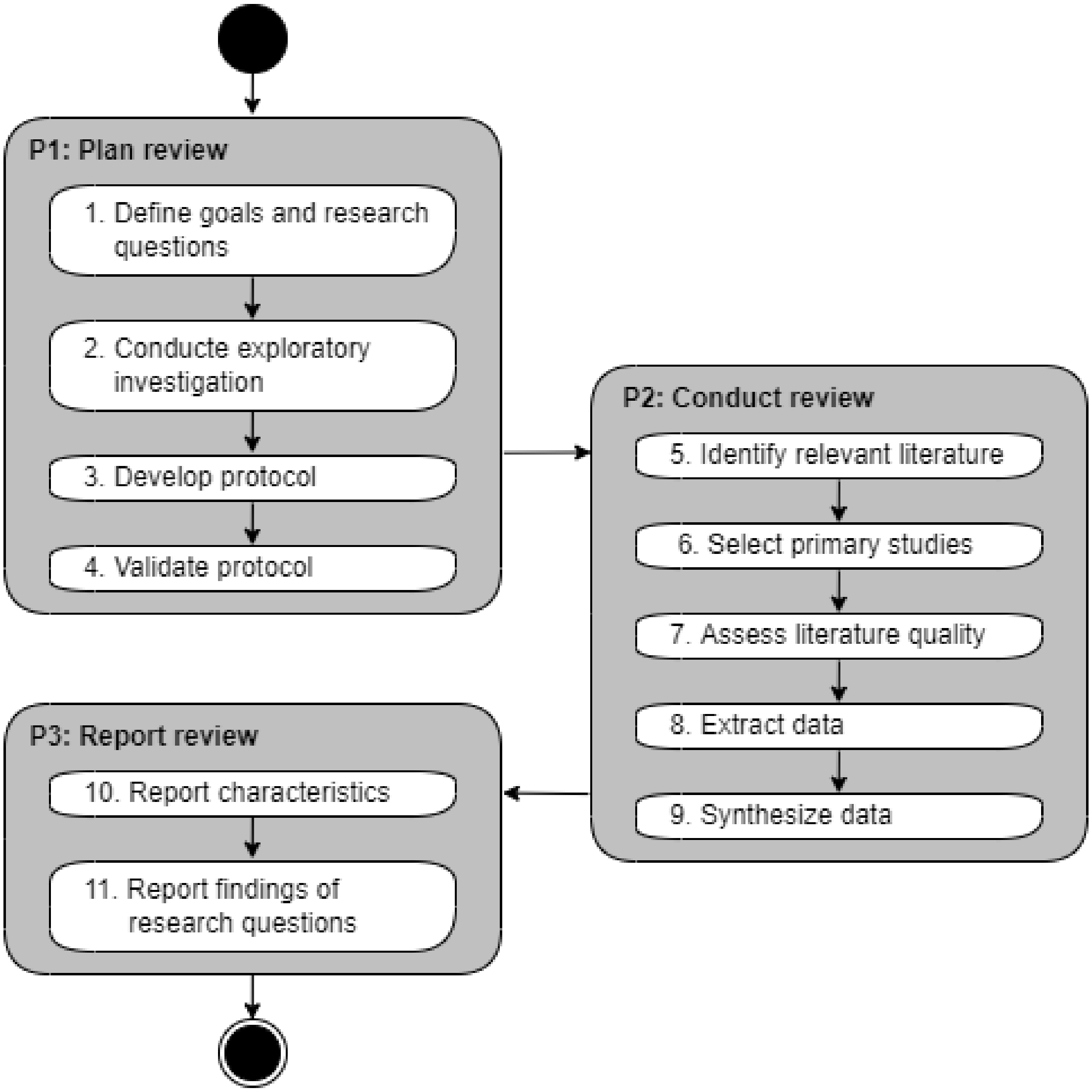
P1: Plan review
1. Define goals and research questions
2. Conducte exploratory investigation
3. Develop protocol
4. Validate protocol
P2: Conduct review
5. Identify relevant literature
6. Select primary studies
7. Assess literature quality
8. Extract data
9. Synthesize data
P3: Report review
10. Report characteristics
11. Report findings of research questions

Figure 3.2: The process side of the PDD for conducting the systematic literature review.

through reading the abstract and conclusion, as well as scanning through the tables. In the seventh step, the quality of the primary papers is assessed by applying a set of quality criteria. Subsequently, relevant data concerning the papers is extracted in relation to the research questions. Thereafter, the data is synthesized, which entails collecting and summarizing the results.

3.1.3 Reporting the review

After conducting the systematic literature review findings in terms of general characteristics are reported. Furthermore, results relating to the goals and research questions are documented. The results of this phase can be found in Chapter 4.

3.2 Collecting expert data

The next research phases involve the collection of expert data, specifically through expert surveys and subsequent interviews. These activities will complement and evaluate the initial set of criteria obtained from the SLR described in Section 4.2.2. Since a lot of literature is conceptual work, industry validation is necessary for practitioners before the extracted SLR concepts can be applied in practice [13].

Participant selection is done using a form of non-probability sampling named purposive sampling [48]. In this case, the qualities of the participant and the corresponding organisation determine the suitability. In particular, an organisation is suitable when it has employed or is planning at least one instance of threat modeling for one or more software system components. Within the organisation, the primary participants must (1) be familiar with threat modeling and (2) have or had a role involving advisory or decision-making responsibilities in the field of IT security. A minimum of eight suitable participants need to be interviewed to obtain 80% saturation [127]. The initial recruitment of organisations is done through email and by using the network of the author. Furthermore, snowballing will be used throughout the interviewing phase to find more participants within a suitable organisation.

3.2.1 Expert surveys

The goal of the surveys is to acquire background knowledge about how the experts tasked with choosing a threat modeling method perceive the issue. This will structure the subsequent interviews in terms of discussion topics. The survey mostly comprises multiple-choice questions, supplemented by a variety of open and drag-and-drop questions. The primary objective is to acquire insights into the current selection approach, with a particular focus on assessing the soundness and comprehensiveness of the quality criteria and requirements identified in the SLR. Moreover, relative importance is tested to comprehend the most fundamental criteria. More details on the contents and structure of the survey can be found in Section 5.1.3.

3.2.2 Expert interviews

The goal of the interviews is to continue evaluating both the soundness and the completeness of the set of requirements and quality criteria. Furthermore, we aim to extract relationships between the qualities and requirements. Moreover, we intend to explore threat modeling method selection from the perspective of the decision-maker, extracting their current approach, the perceived goals of threat modeling, the perceived scope of threat modeling, and domain preferences. This is accomplished by considering the responses from the pre-interview survey as the foundation for determining relevant topics. To be able to capture industry insights and achieve the goal at the same time, semi-structured interviews with experts across multiple organisations are done. The duration of these interviews typically ranges between 45 to 60 minutes, and if necessary, a follow-up interview can be requested. The interviews will take place in person at the location of preference of the interviewee or can be held online and are recorded when consent is obtained from the interviewee. The sole purpose of the recording is to post-process the interview into reportable results. This way, the interviewer can focus on the interview rather than taking notes.

Preparing the interviews helps maintain a necessary structure for validating the list while providing room for exploring the domain of organisational decision-making criteria. Hence, the interview questions are prepared beforehand and are designed to change the flow of the conversation when necessary. The prepared questions are part of the interview guide, which the interviewer can consult. Besides prepared questions, this guide also contains subjects to explore and a rough timetable to guarantee a smooth course.

The contents of the interview are divided into six main parts. It starts with an introductory segment in which the interviewer introduces himself and provides information about the goal of the interview. In addition, the interviewee is asked to read through and sign a consent form. Besides disclosing the interview's contents, the interviewee is informed of their voluntary participation, including the option to withdraw at any time. It is also stated that the data extracted from the interviews will not be bound to any identifiable participant. In addition, a second signature is asked to provide consent for audio recording. Afterward, a new phase starts in which the validity of participation is verified. The interviewee is encouraged to talk about what threat modeling means to them and how they are involved with threat modeling within their organisation. If the participant appears to be missing any of the previously stated required qualities, the interview will directly continue to the final phase.

The exploratory part (Part 3) should be instantiated if the participant has proven to be suitable. The primary objective of this phase is to gather information about the practical approach to threat modeling method selection and to identify the factors that influence this decision. Also, we want to extract the factors that might contribute to their perspective such as the perceived goal and scope of threat modeling, as well as the importance of the tool and technique components. During this phase, the interviewer evaluates potential correspondence between the elements mentioned and criteria found in the literature. In the fourth part, the interviewer takes the lead and asks questions regarding the survey answers, aiming to elicit contextual insights related to the selection of the answers. In the second to last step, the goal is to elicit relationships between qualities and requirements, by posing questions regarding the perceived connection between them. Once the interview ends or the last ten minutes are entered, the interview progresses to the final phase. The participant is thanked and asked for their final input. Furthermore, this phase involves asking about their willingness to participate in a follow-up interview and potentially assisting in finding additional participants.

It is noteworthy to mention that several challenges were encountered with these interviews, especially concerning the allocated time and content presented. These have been dealt with by reformulating the requirements found in the SLR to user stories, introducing the pre-interview survey, and splitting up the contents of the interview over multiple participants. Detailed information on the specific process and outcomes of this can be found in Chapter 5.

3.3 Designing the systematic selection approach

The systematic selection approach is centered around a decision model. This decision model is adapted from the framework of Farshidi [51]. The procedure of going from a decision-making method to this framework is elaborated on in Section 2.2. In this section, we aim to explain how the framework is utilized and adapted to create the decision model. The final design of the approach can be found in Section 8.4

The framework utilizes the six phases of Majumder [108]: (1) objective identification, (2) feature selection, (3) alternative selection, (4) weighing method selection, (5) applying the method of aggregation, and (6) decision-making based on the aggregation results. The general contents of these phases can be found in Section 2.2.2.

The domain objective serves as input for the first phase and is similar to the goal of this study. This study aims to create a systematic decision-making method for threat modeling method selection. Therefore, the corresponding domain objective is to perform threat modeling method selection. To understand the domain and its qualities, this research abstracts the threat modeling concept using the method engineering methodology [17]. Threat modeling can be defined as an activity as part of a more extensive system development process. Such an activity contains a set of sub-activities, which are described *tasks* resulting in a *deliverable*, utilizing some *technique*, and can be supported by some tool [17]. Studying the relationship between definitions resulted in the observation that a task is related to a deliverable and a technique corresponds with the tool. A task and deliverable can differ given a specific situation or case. However, tools and techniques seem to have a general application. Therefore, the quality model of Farshidi [51] should be adapted to the context of a threat modeling method. This is achieved by adapting the method internal quality criteria [19] to the context of a threat modeling method and using them as quality groups. These groups are populated by quality criteria that are extracted from literature, expert surveys and interviews.

The feature selection is performed by choosing a subset of threat modeling requirements, extracted and refined in the systematic literature review, expert surveys and interviews. These requirements can pertain to a threat modeling technique, tool, notation, or a combination of these components. Given the outcomes of the expert interviews and the domain specification of the quality criteria, a mapping between the quality criteria and decision-making criteria can be established [51]. This process is facilitated by input from domain experts.

Alternative selection is grounded in the systematic literature review by backward snowballing on existing literature reviews and other comparison studies regarding threat modeling. In the context of this study, alternatives are defined as a combination of a threat modeling technique utilizing a notation, which may be supported by a tool. If multiple tools support a method, various alternatives should be considered. Furthermore, a mapping between the alternatives and decision-making criteria must be established. Due to resource constraints, we opted to directly map the requirements to the alternatives. Although a quality requirement mapping is constructed, future work is invited to validate this mapping and consider the quality dimension as proposed by Farshidi [51]. A Boolean value is assigned to each alternative-requirement relationship to indicate the application of requirements to an alternative. Additionally, partial inclusion is considered for process-dependent requirements. These relations are established given method literature, tool documentation, and consulting a domain expert.

Weighing method selection, as the name suggests, is occupied with selecting the weighing method to extract the importance of criteria from stakeholders. Qualitative weights are used. Numerous qualitative approaches use an ordinal scale such as a 5-point Likert scale to determine the weights [35, 53, 106]. Farshidi [51] proposes to use MoSCoW to extract the subjective preferences of decision makers [51]. MoSCoW stands for [174]:

M - This criterion must be met to succeed.

S - Should have this criterion if feasible, but not critical for success.

C - Could have this criterion because the stakeholders welcomes it. However, implementing such criteria may not affect anything else.

W - Won't have this criterion in the project's current version.

During this research, it is suggested to use MoSCow for both the requirements and method criteria, since decision-makers already prioritize their domain requirements based on MoSCoW [32].

The prioritization is obtained through consulting the decision maker(s). When there are multiple decision-makers, each is asked to prioritize every criterion. Then the data is compared for inconsistencies. For example, when Expert 1 (E1) prioritizes Criteria 1 (C1) as *Should have* and Expert 2 (E2) prioritizes the same criteria as *Won't have*, there is an inconsistency. A session is then initialized in which E1 and E2 need to discuss their opinion, decide on the inconsistent prioritization, and reach an agreement.

The aggregation method is based on the weighted sum. Hard constraints are enforced on requirements with a *Must have* and a *Won't have* priority. Whenever these constraints are violated by the alternative, the method becomes infeasible. Numerical values are assigned to each MoSCoW priority, based on the budget distribution, so that they can be used as weights in the calculation [120]. A score is calculated for each feasible alternative, based on the requirement prioritization. Moreover, the alternatives that comply with the hard constraints are then ranked in descending order.

The systematic approach utilizes an instance of this decision model and adds a preliminary activity to guide the practitioner through the prioritization process. This establishes a dynamic prioritization process grounded in the perceived goal and scope of threat modeling. Furthermore, an experimental global multiplier is incorporated to deal with requirements that are perceived as important but require context beyond the method itself. The details of the decision-making approach and decision model can be found in Chapter 8.

3.4 Case study

In the last phase of the research method, the proposed systematic decision-making method will be tested through evaluation in practice. The objective of the evaluation is to determine whether the systematic decision-making approach contributes to a traceable and objective selection of a threat modeling method within the context of a large enterprise, and to pinpoint points of improvement. The approach is evaluated in terms of effectiveness, accuracy, usefulness, ease of use, completeness, and functionality.

The case study is performed within a.s.r., a large insurance company in the Netherlands. Currently, their teams are tasked with creating a threat model as part of a checklist when employing a service in the cloud. This is done using the STRIDE model of threats and the Microsoft Threat Modeling Tool (TMT). For every project, a threat model is created by a team and evaluated by a security officer.

The systematic selection approach is enrolled for a group of security experts, responsible for giving advice on security matters. After the approach is tested and the ranked list of alternatives is obtained, the participants of the case study are asked to fill in a post-task questionnaire. This questionnaire comprises a mix of positive and negative statements, which participant rate based on their level of agreement. Each criterion is represented by two to three statements. These criteria are a subset of the Prat taxonomy. The Prat taxonomy is widely used to help select evaluation criteria for information science artifacts.

4 Systematic literature review

The systematic literature review aims to create a foundation for the proposed decision-making method. Section 4.4 contains the identification of comparison research related to threat modeling. In addition, subjects for comparison are extracted in order to get an overview of the comparison scope of threat modeling methods. This provides an answer to SRQ1:

How can existing threat modeling approaches be compared?

Moreover, Section 4.7 discusses the utilization of criteria related to threat modeling selection. This addresses SRQ2:

What are the criteria for selecting threat modeling approaches?

The decision-making criteria and alternatives identified in this chapter form a foundation for the decision model constructed in SRQ3:

What type of systematic approach can be used for threat modeling method selection?

4.1 Previous literature reviews

Several literature reviews have been conducted to map the existing threat modeling landscape, and at least three of them have been published in the last five years [159, 175, 200]. Among these, two were executed systematically. As explained in Section 3.1, the SLR performed in this study is conducted based on this set of reviews. In this section, we provide an overview of the preceding literature reviews upon which our systematic literature review is constructed.

Tuma et al. [175] did an SLR that focused on threat analysis in software systems. Two search queries were released on three different databases, these queries slightly varied based on the positioning of the asterisks (*). Additionally, snowballing was performed. In total 26 approaches were identified based on 38 papers published between 1998 to 2016. These approaches were assessed and compared based on criteria related to the applicability, required input for the analysis, the analysis procedure, the outcome of the analysis, and ease of adoption. Besides providing a detailed overview of the 26 approaches and their characteristics, the study concludes that the approaches lack quality assurance of the outcome, maturity, validation, and tool support.

A SLR by Xiong et al. [200] was carried out to investigate the concept of threat modeling and to gain insight into the state-of-the-art developments in the field. Four different electronic libraries were used to identify 176 unique papers that were published between the years 2004 and 2017. From this initial search, 54 articles were classified into three different classes and individually analyzed over a variety of criteria. The criteria regarded the type of method, the system analyzed, threats covered, and the validation method. Furthermore, Xiong et al [200] summarized the focus and approach of the methods based on multiple criteria. The findings indicate a notable gap in the definition of threat modeling methods. Furthermore, state-of-the-art work in the field mostly involves manual processes and exhibits flexibility regarding the selected approach [200].

Shevchenko et al. [159] created a technical report that provides a summary of available threat modeling methods. How the methods were selected was not specified. Each method was described in terms of technique and notation used to perform the method. Also, the context in which the methods could be applied was occasionally mentioned. This summary was based on a mix of published papers and official documentation. The report concluded with a comparison summary based on the unique features of each method. In addition, certain contextual factors were provided that could be utilized when choosing a threat modeling method

4.2 SLR Approach

The actual SLR was conducted, after planning the SLR in Section 3.1 and obtaining the background knowledge regarding previous SLRs. In this section, we provide details on how the actual SLR was performed. Describing each step and providing the specifications utilized will contribute to the traceability of this study. The subsequent paragraphs will address the search terms, source, criteria, and data extraction used in this study.

4.2.1 Search terms

It has been hypothesized that there is no common structured protocol to compare or select threat modeling methods. To test this hypothesis, a database search on Scopus is executed to create an initial dataset. Both Tuma et al. [175] and Xiong et al. [200] utilized a systematic search query to obtain the initial set of threat modeling literature. This research combines both queries and extends them with the following keywords: *method* selection, method* comparison, approach selection, approach comparison, systematic literature review, taxonomy of threat* and threat* taxonomy*. By using the asterisk, common language differences are mitigated. Additionally, a wider range of results is collected as a result of including terms that are frequently utilized interchangeably in the literature, such as method(ology). The resulting query can be found below.

```
( cyber OR network OR IT OR ICT OR information ) AND
( secur* OR privacy OR abuse OR misuse OR risk OR threat* OR attack*
OR flaw* ) AND
( analysis OR assess* OR model* OR management OR elicit* ) AND
( "taxonomy of threat*" OR "threat* taxonomy" OR "method* selection"
OR "method* comparison" OR "approach selection" OR "approach
comparison" OR "systematic literature review" )
```

In addition, several acceptance criteria were used to further refine the search and narrow the scope of the results. While executing the search query, the following filters were applied in Scopus:

1. *Subject area limited to computer science; social science; decision-making; business, management and accounting).*
2. *Document type limited to journal articles or conference papers.*
3. *Publication year beyond 2009.*
4. *Language limited to English.*

4.2.2 Data extraction and acceptance criteria

The search string and its filters provided 940 results (December 2022). This list was narrowed down by manually scanning the abstract, title, keywords, tables, and figures and applying the inclusion and exclusion criteria from Table 4.1. It was decided to scan tables and figures since we observed that one of the primary papers did not mention anything about criteria inside the abstract, keywords, or title. However, the paper provided a recommendation for stakeholders based on a list of features [159], which is also relevant for answering SRQ2.

Table 4.1: Inclusion and exclusion criteria.

Inclusion criteria
1. Studies that compare methods, techniques, or tools for identifying and analyzing cyber security threats or attacks.
2. Studies that provide an overview of the cyber security threat modeling landscape.
3. Studies that mention criteria, characteristics, or features that apply to (components of) threat modeling.
4. Studies that include a recommendation for practitioners or stakeholders regarding threat modeling.
5. Case studies that evaluate a threat modeling tool, method, or technique in a business context.
6. Studies that include techniques, methods, methodologies, or tools for identifying and analyzing security threats for a software system (subject of comparison).
Exclusion criteria
1. Studies for which the full text is not available.
2. Duplicate studies.
3. Studies that solely regard physical security threats.
4. Publications about safety-hazard analysis, anomaly, or intrusion detection.
5. Publications focusing on concrete instances of security threat taxonomies, threat models, or security analysis for a specific case.

Scanning the documents and applying the criteria identified thirteen additional papers. Therefore, backward snowballing is employed to extend the relevant dataset. This type of snowballing is done according to the framework of Wohlin et al. [197]. Backward snowballing, accomplished by utilizing the reference lists of the relevant papers, resulted in 968 additional references which further helped identify methods, techniques, and tools. Even though this is not the primary goal of the systematic literature review, obtaining a set of subjects and components that can be compared is necessary for creating the decision model, which is a core part of the systematic selection approach.

After the same inclusion and exclusion criteria are applied to the literature obtained through snowballing, the papers were checked for relevancy to the research questions, and quality criteria were applied. This involved the exclusion of poorly cited results (<10 citations and over five years published). Also, newly found papers regarding model-driven security engineering approaches and automatic vulnerability detection methods were disregarded. Although model-driven security engineering is very related to threat modeling, the main purpose is to integrate non-functional requirements in development rather than identifying and assessing threats. Automatic vulnerability detection is removed from the scope due to its reactive nature. Meaning, it typically reveals vulnerabilities after the code is created. Threat modeling is proactive and should be employed before this development phase. This resulted in a final data set of 127 papers. Figure 4.1 shows an overview of the process.

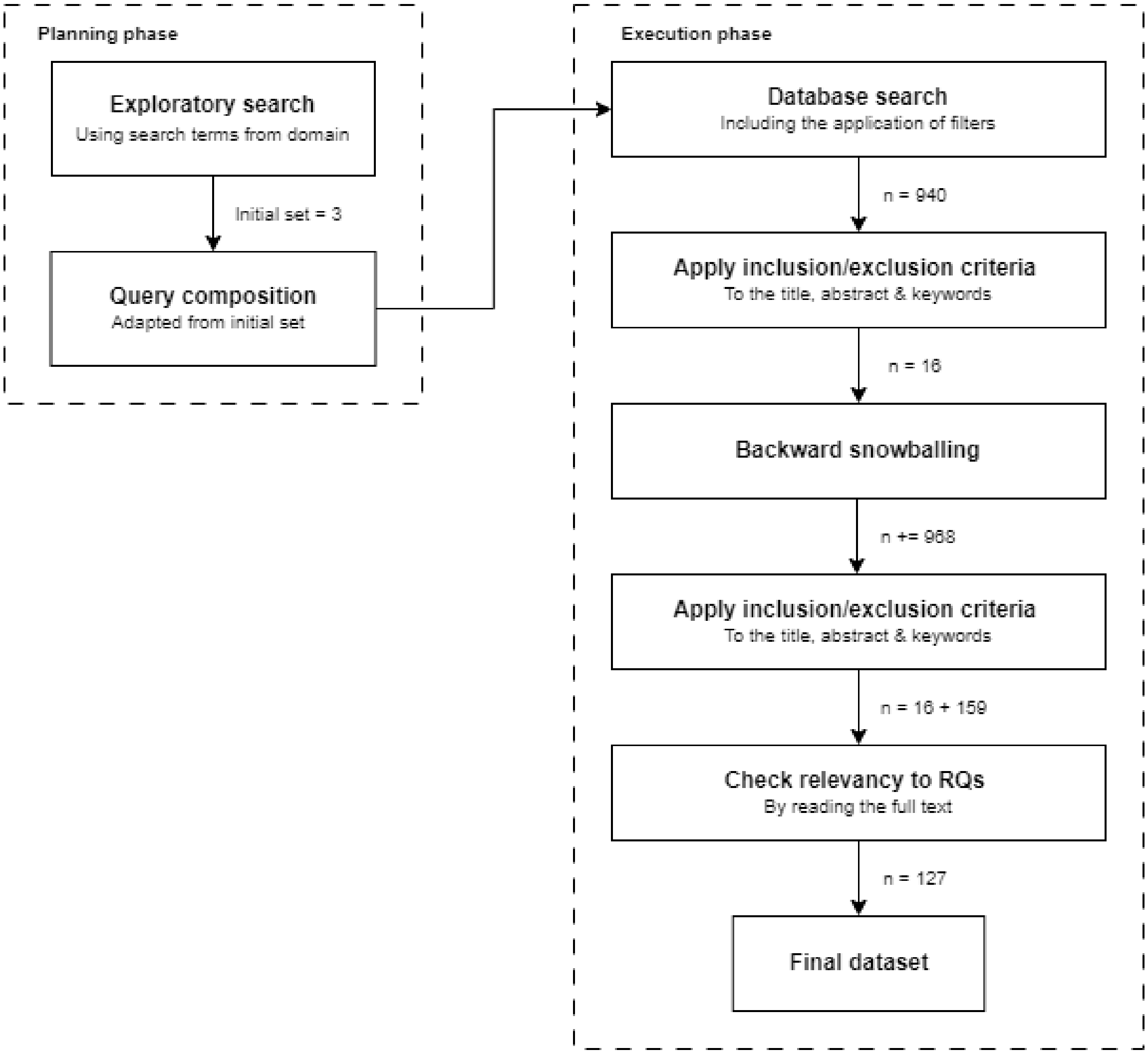

FIGURE 4.1: The planning and execution phase of the systematic literature review. The number of papers is indicates as "n".

4.2.3 Data extraction

The SLR contributes to multiple research questions by doing one data extraction process. Therefore, it can be confusing what data was harvested. This section aims to provide an elucidation of the data gathered from the 127 papers, which were selected as the final dataset. The papers were classified based on their contribution to answering the research questions. The focus of SRQ1 is to harvest papers that compare threat modeling and threat modeling-related concepts. Furthermore, the subjects of comparison, such as tools, techniques, and methods are classified as alternatives and were obtained to provide a foundation for the decision model (See Chapter 8 for more details). To contribute to SRQ2, papers that contain comparison criteria in terms of qualities and requirements are extracted. Several papers that conducted a comparison also contained suitable alternatives. Meaning that there exists a certain degree of overlap between the classes. Figure 4.2 shows the classification of the final dataset and their corresponding harvest. The following sections contain the findings of the SLR, structured based on this classification.

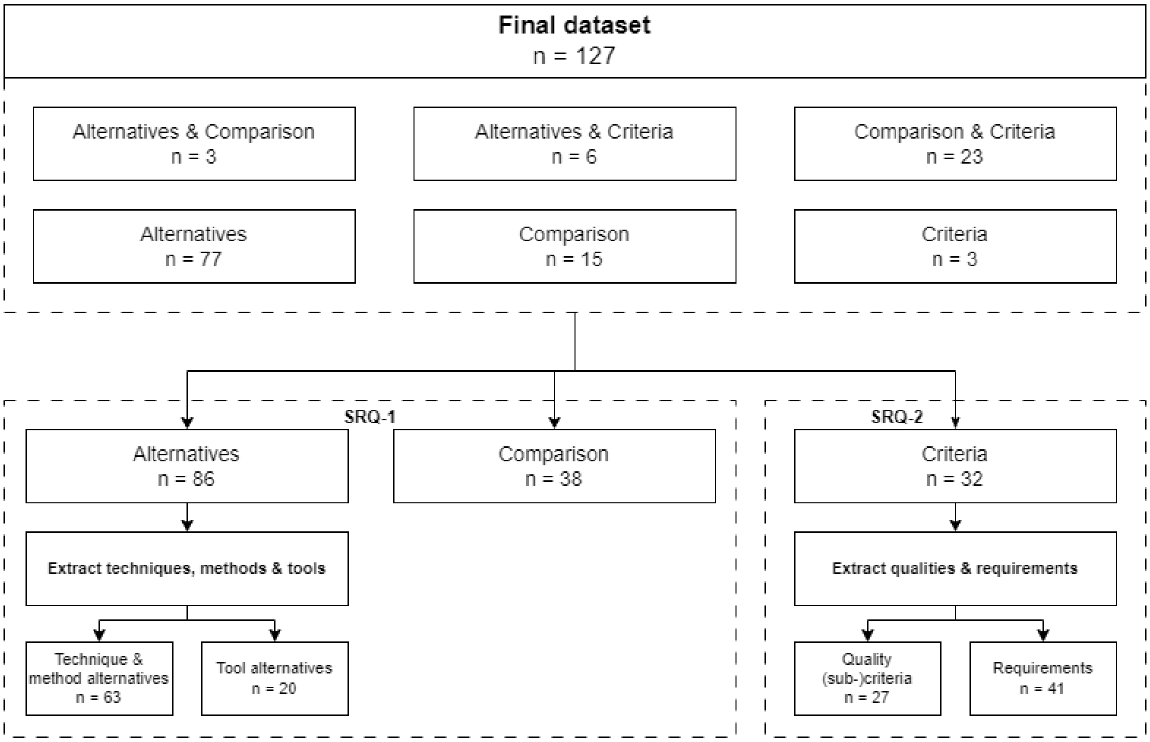

FIGURE 4.2: The classification of the papers from the final dataset and their relevance in addressing SRQ1 and SRQ2.

4.3 General characteristics

Due to filters not being applied to the results from backward snowballing, the studies were published between 1998 and 2022. Figure 4.3 shows a timeline of the 127 papers included in this SLR. In general, the research interest in the field of threat modeling appears to be relatively consistent, averaging about 5 publications each year. The all-time peak of interest was in 2013 with 12 published papers in one year. A slight upwards trend in the number of publications can be observed over the years, starting around 2000 when the publication count was at its lowest point up until the present time.

As depicted in Figure 4.2 there are six different classes of papers identified, resulting in three different categories. The majority of papers contained (a part of) an alternative. This means the paper explained or used at least one threat modeling method, technique or tool. Although notations are also considered to be a part of a threat modeling method, these come together with a specific technique and therefore were not explicitly considered in the SLR. Certain papers were classified as containing an alternative but also were doing a comparison. In this case, there was no clear use of common criteria in order to compare the alternatives. The comparison category was assigned to papers that compared security-related methods, which included not only threat modeling methods but also other security approaches. There is a gap in the literature for the direct comparison of threat modeling methods, this required the SLR to gather information beyond the threat modeling domain. Furthermore, a paper was categorized with the criteria tag when there were common comparison criteria, which potentially could be used in the context of selecting threat modeling methods. Each of the three clusters of papers will contribute towards creating the foundation for answering (a part) of SRQ1 or SRQ2. Each cluster was analyzed, and the results can be found in the following sections.

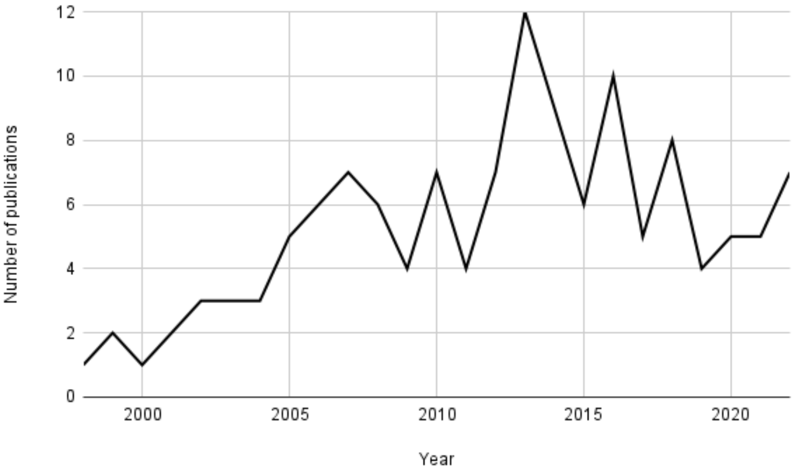

Figure 4.3: Year of publication for the selected papers.

4.4 Comparing security related methods

A wide variety of papers that compare threat modeling and security-related methods are identified. Out of the 38 selected papers, eleven papers are specific to the topic of threat modeling. Among these papers, two specifically focus on comparing and discussing threat modeling methods and utilize predefined criteria in a general application context [175, 200]. These are equal to the systematic literature reviews mentioned in Section 4.1. This confirms there is a gap in general comparisons for threat modeling. In two other instances, threat modeling methods and techniques are summarized, shortly evaluated, and compared based on qualitative aspects. Shevchenko et al. [159] emphasize distinctive attributes that practitioners can base their selection on. While Selin et al. [156] authored a thesis that evaluated the maturity and suitability of threat modeling methods. Their conclusion suggests that none of the assessed approaches were mature enough, indicating a need for their combination to cover all spectra [156]. Furthermore, qualitative advantages and disadvantages of the methods were presented.

Also, threat modeling approaches are assessed for a specific application. Tatam et al. [168] compares multiple approaches within the context of Advanced Persistent Threat (APT) style attacks, by offering an overview of advantages and disadvantages. Additionally, the authors propose a preliminary set of features that can be employed in the selection process [168]. Another study conducted by Wright et al. [199] carries out a systematic review in which threat modeling approaches utilizing Bayesian networks are contrasted within the application context of smart city infrastructure. This is accomplished based on a set of criteria that primarily describe the structural composition of the associated papers. Scandariato et al. [150] aim to provide a descriptive evaluation of STRIDE, using several quantitative benchmarks. They explain that other studies typically evaluate threat modeling methods through a comparative analysis of two or more approaches [150].

The remaining papers are devoted to comparing an assortment of threat modeling tools. Shi et al. [161] developed an evaluation taxonomy to assess and compare threat modeling tools, and subsequently applied the taxonomy to compare six of the most commonly used tools. The taxonomy consists of Boolean and textual comparison criteria. Bygdas et al. [22] compared and evaluated Microsoft TMT and OWASP Threat Dragon. This process is conducted through a qualitative assessment, involving an evaluation of factors such as the intention of usage. Lastly, Granata et al. [60] compared three open-source threat modeling tools based on a specific application scenario. However, the use of the specific application scenario leads to a generalization challenge.

4.4.1 Comparing partly related methods

Due to the lack of a common threat modeling definition, ten papers were extracted that compare threat modeling approaches as part of a broader concept. In six of the eleven papers threat modeling was associated with Requirement Engineering (RE). Several of the subjects examined in these studies, such as misuse case, CORAS, and abuse case, have been classified before as threat modeling techniques [175]. Daramola et al. [34] conduct a comparison between I* based and use case-based security requirement engineering approaches using their internally developed framework called the Security-oriented Modeling Approaches Characterization Framework (SMACF). This framework primarily consists of criteria with Boolean values, as well as a limited number of qualities that can be assessed with high, medium, or low. Munante et al. [125] review Security Requirement Engineering (SRE) methods to analyze which ones are compatible with risk analysis processes. A set of criteria that encompass integration have been used to achieve this. Also, a general comparison has been conducted based on ISO27005.

Elahi et al. [45] introduce a framework for the selection and analysis of security requirements. Furthermore, the proposed framework is compared with four asset and threat modeling methods. Each method is assessed against each criterion and evaluated based on perceived consideration, which may be explicit, not considered, or implicitly considered. Both Fabian et al. [49] and Mellado et al. [117] compare security requirement engineering methods based on a set of criteria that encompass multiple sub-criteria. The application of the sub-criteria can be assessed as either a full match, no match, or partial inclusion. Finally, a study utilized criteria related to the common tasks of the approaches to compare SRE methods, based on their coverage of specific activities [149].

Two papers associate threat modeling with risk assessment. Raspotnig and Opdahl [140] compare risk identification techniques using eleven distinct criteria. The authors evaluate each selected approach through a textual assessment. The results assist practitioners in selecting and combining a variety of risk identification techniques. On the other hand, Maheshwari and Prasanna [107] explain how the integration of threat modeling and risk assessment into the SDLC could lead to a greater understanding of security risk. As a byproduct, threat modeling techniques are compared based on their goal and security properties.

The remaining papers do not share any commonalities; however, a concise discussion is presented regarding their comparison context. Lagerström et al. [98] conduct a systematic literature review on threat modeling and attack simulations of smart cities. In this review, threat modeling and attack graph-based methods are implicitly compared based on their main contribution. Another systematic review focuses on web application security development models. Out of the 24 identified security development models sixteen of them used threat modeling as their approach. In addition to assessing the models based on their approach, the paper identified the security techniques used for vulnerability mitigation [126]. Hussain et al. [76] review approaches to model security of software systems, which contain threat modeling and other types of model-driven methods. Instead of evaluating these methods using a predefined set of criteria, each approach has been summarized in a paragraph.

Another high-level comparison is reported in the study of Uzunov et al. [179]. This study offers an extensive comparison of all sorts of security methodologies or methods including threat modeling approaches. For each item, the paper includes an outline, an elaborate description, and a discussion. The discussion pertains to the industry adoption of a security methodology. Lastly, Farah et al. [50] uses generalized criteria to classify and compare approaches for the security of business processes. This study includes several threat modeling methods and techniques as subjects of their comparison.

4.4.2 Comparing other security-related methods

The remaining fourteen papers did not specifically contain threat modeling methods but conducted a form of comparison within a domain encompassing a cyber security method. Six of these papers were dedicated to comparing risk assessment and risk management methods. Marle and Gidel [111] specifically focused on aiding method selection, for which an MCDM framework was employed to support the selection of risk management methods. The selection criteria used in the decision model consisted of three to five levels. For each level, a corresponding description of its application is provided. Furthermore, three studies developed a general taxonomy to facilitate the comparison of methods. [29, 157, 158]. The final two studies were devoted to comparing risk assessment methods based on completeness, so comparison criteria were utilized that relate to the general tasks of the methods [189, 190].

In the overarching domain of model-driven security, the majority of identified papers did a systematic literature review of the existing methods and notations. Each paper utilized a different set of criteria in order to classify and compare their subjects. These can be both qualitative [167], for example, based on usability, as well as, quantitative (feature-based) [12, 57, 130]. Moreover, one study reported a comprehensive literature summary in which approaches were described according to a set of predefined topics [103].

Additionally, a study was identified that conducted a comparison by performing a systematic literature review on vulnerability detection systems. This was done based on features, development phases, and the underlying method [142]. Another study focused on the topic of security assurance, wherein their proposed model was compared to alternatives based on their goals [88]. Finally, Villarroel et al. [187] conducted a literature review on methods for secure information system development in which the methods were assessed based on a set of technique and specification-related quality criteria.

4.5 Answering SRQ1

The literature suggests there is no standardized approach for comparing threat modeling or related methods. There are only a few studies specifically focusing on comparing within the domain of threat modeling. Upon analyzing these studies, three common comparison approaches emerged. Firstly, the comparison is most commonly based on criteria that relate to the specific goals and contexts of the study. These criteria may be open-ended, meaning they are described using text, or organized into categories to assess the inclusion of the methods. Certain studies introduce a (implicit) taxonomy that, for example, combines Boolean and categorical criteria to display the inclusion, providing a more structured comparison. Secondly, certain studies compare threat modeling methods and their components based on their strengths and weaknesses. This is most often a semi-structured assessment, guided by the author's qualitative interpretations. Lastly, certain studies provide a summary for each method and conclude with a brief evaluation of the respective approaches. Additionally, a limited number of studies were identified that focused on comparing threat modeling tools. A limitation observed in the majority of identified studies is their narrow focus because they focus on a limited number of subjects or use a restricted set of criteria for comparison.

Upon further examination of studies that include threat modeling approaches as a part of their study or that compare related topics, additional instances of these comparison approaches were found. In more mature domains, such as risk assessment or security requirement engineering, the inclusion of systematic literature as part of the comparison process is more common. These studies often use a structured taxonomy, consisting of both requirements and quality criteria. Criteria that assess the inclusion in Boolean values are also more prevalent, and partial inclusion is commonly considered for such criteria. Furthermore, certain studies compare methods based on criteria related to the most generic tasks of the method domain.

4.6 Alternatives

Most papers in the SLR contained information about an alternative threat modeling method, technique, or tool. A total of 86 papers were considered to identify both methods, techniques, and tools. It was decided to group the methods and techniques together since the boundaries between them are not clearly defined within the papers. These terms are interchangeably used together with "methodology". The identified alternatives will create a basis for the decision model.

4.6.1 Methods and techniques

Examining the papers that were marked to contain an alternative resulted in identifying 63 unique potential methods and techniques. Due to the absence of a universally agreed-upon definition for threat modeling [200], it was decided to adopt the broadest possible scope for the extraction process. As a result, there are cases where methods encompass multiple techniques or integrate threat modeling within a broader method, potentially encompassing areas like risk assessment. In other instances, threat modeling is implicitly performed but referred to using different terminologies. The diverse range of identified candidates will serve as the foundation for the final set of alternatives in the decision model. The process of selecting and constructing these alternatives is elaborated on in Section 8.3.1.

Table 4.2 shows an overview of all methods and techniques identified and their associated papers. Additionally, each candidate is summarized based on their main characteristics and unique focus points. This provides an overview of the current threat modeling landscape. Moreover, tool support indicates the availability of a tool for the respective method. The term *"Not-specific"* refers to cases where the tool associated with the candidate is a non-specific drawing tool or a combination of multiple tools that are associated with the existing threat modeling techniques the method utilizes. Ten of the threat modeling methods (16%) utilize this type of tool. Twelve methods (19%) have a *"prototype tool"*, which indicates that the paper mentions the existence of a tool, but it is not publicly released. If no tool is specified, it is indicated with *"no tool"*, otherwise the name is mentioned. No tool was specified for eighteen methods (29%), while nineteen (30%) had a specific tool. Note, that these are not an exhaustive list of identified tools and may also contain tools that are discontinued. Further details on the identification of tool alternatives are found in the next section.

Table 4.2: An overview of all threat modeling methods identified in the SLR. A short summary of the main contents is provided.

Method / technique	**Ref**	**Content summary**	**Tool support**
Abuse cases	[115]	Adapt use cases to capture and analyze security requirements, based on external threats actors.	Non-specific
AEGIS	[54]	UML meta-model of the definition and the reasoning over the system's assets.	No tool
Attack graphs	[160]	Automated generation and analysis of attack graphs based on symbolic model checking.	Prototype
Attack paths	[28]	Automated quantitative threat modeling, based on attack path analysis.	Tiramisu
Attack scenarios	[3]	Attack scenario detection in source code, based on formal signatures.	Prototype
Attack trees	[113, 146, 155]	Analyze possible attacks on a system, in a hierarchical manner.	SecurITree
Attack-defense trees	[93]	Extending attack trees by allowing nodes that represent defensive measures.	Prototype
Attacker agents & goals (I*)	[44, 102, 123, 124]	Agent oriented threat modeling with goals.	i* model tools
Automated analysis of security-design models	[82]	Verification framework supporting the construction of automated threat analysis tools for UML diagrams	Plug-in for CASE
Automated attack trees	[100]	Automated generation of attack trees.	Prototype
AutSEC	[56]	Identifcation and mitigations trees to conduct automated threat modeling.	Prototype
Center of Gravity	[165]	Threat modeling using attack paths, focused on most critical assets.	No tool
CORAS	[39, 105]	Risk-based threat analysis, using treatment diagrams and descriptions.	CORAS Web Tool
ESSecA	[138]	Expert System for Security Assessment that guides threat assessment of IOT systems by producing a threat model.	Code in paper
Executable MUC	[194]	Formal specifcations of misuse case scenarios, can be executed with a misuse case model to identify mitigations.	MUCSIM
Extended CPTM	[58]	Extension of Cloud Privacy Threat Modeling (CPTM) method for perserving privacy throughout development.	No tool
Fault-tree analysis	[66]	Uses fault tree to illustrate the logical relationships between events that can lead to a security breach. The paper attached uses a fuzzy version in combination with a mathematical model to calculate risk.	prototype
HARM	[86]	Combines Attack Secuence Descriptions, Misuse sequence diagrams, Misuse case maps, Misuse Case, Attack trees, and Attack Patterns.	Non-specific

hTMM	[116]	Combines Security Quality Requirements Engineering Method (SQUARE), Security Cards, and PnG activities.	Non-specific
Insider threats	[83, 84]	Illustrate and classify insider threats, based on formal modeling and analysis of infrastructures of organisations.	Code in paper
IoTRiskAnaly	[122]	Automated threat model generation, risk-based analysis and prioritization for IoT systems.	PRISM
KAOS	[70, 182, 183, 184]	Threat graphs, anti-goals, anti-models, and threat trees.	Non-specific
Kong-Threats	[92]	Threat analysis through simulating state transitions, using misuse cases and STRIDE.	Prototype
LINDDUNN	[37]	Threat to (DFD) element mapping, privacy modeling.	LINDDUN GO
MisUse Case Maps (MUCM)	[87]	Maps threat actor behaviour to the architectural context of a system.	Non-specific
Misuse cases (MUC)	[163]	Eliciting security requirements based on use cases that focus on internal misuse.	Non-specific
Misuse patterns	[47]	Identifying security threats by leveraging known patterns of misuse or abuse of system functionality. Attached paper focuses on cloud applilcations.	Non-specific
Multi-attacker threat model	[5]	Allows each internal actor to behave maliciously, through interception and forging messages. Embeds automation through use of an existing model checker.	Code in paper
NEMESIS	[85]	Automated architecture for threat modeling and risk assessment.	Prototype
OCTAVE	[2]	Risk-based threat assessement, reliant on human interaction, focuses on organisational risk.	No tool
OCTAVE Allegro	[23]	Alternative version of OCTAVE with the focus on information assests and their context.	No tool
P.A.S.T.A.	[177]	Threat scenarios with associated risk and countermeasures, using attack trees and DFD.	Non-specific
PERSONA NON GRATA	[30]	Characterize users as archetypes that can misuse the system and view the system from the point of view of unintended use.	No tool
Petri-nets	[201]	Model inteded behavior of system and security threats using petri-nets.	Non-specific
Problem frames	[9, 71]	Problem-oriented security patterns, which can be re-used to identify problems on different abstraction levels.	UML4PF, etc.
PwnPr3d	[79, 80]	An attack-graph-driven probabilistic threat-modeling approach	Prototype
Qualitative TM method	[136]	Combines Attack trees, STRIDE, and CVSS vulnerability scoring.	Prototype
Quantitative TM for cloud	[154]	Stochastic risk-based threat analysis, that helps select security controls that maximize return on investment.	Prototype
Rule-based graph matching	[11]	Automatically extracting threats, using threat catalogs and extended DFD.	Code in paper

Secuirty cards	[116]	Emphasizes on creativitiy and brainstorming, based on content presented in card format.	No tool
State charts	[43]	Modeling behavior of attackers using state charts.	non-specific modeling tool
STORE	[4]	Security Threat Oriented Requirements Engineering Methodology, that utilizes STRIDE to identify threats.	No tool
STRIDE	[150, 172]	Threat identification based on Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service (DoS), and Elevation of Privilege.	Micosoft TMT
Trike	[147]	Open-source risk-based threat modeling method from a defensive perspective, focusing on coordination and collaboration.	Octotrike
Unnamed by Abe et al.	[1]	Map threat patterns to business process, represented in sequence diagrams, to derive negative scenarios.	Non-specific
Unnamed by Beckers et al.	[8]	Security pattern based threat modeling that integrates multiple modeling methods such as CORAS and Misuse cases.	UML4PF
Unnamed by Bedi et al.	[10]	Combines several existing techniques to create a three-phased risk-based threat modeling and risk management approach.	Prototype
Unnamed by Burmester et al.	[20]	Threat modeling for Cyber-physical system, requirements represented as security policies are used to identify system vulnerabilities.	No tool
Unnamed by Casola et al.	[24]	Semi-automated threat modeling assigning threats based on assets using a STRIDE threat library, and countermeasures from NIST.	SlaGenerator
Unnamed by Cerotti	[25, 26]	Automated threat modeling using MITRE frameowkr for smart energy grids.	No tool
Unnamed by Elmrabit	[46]	Probablistic threat modeling focused on insider threats based on Bayesian networks.	No tool
Unnamed by Granata and Rak	[59]	Automated threat modeling and risk evaluation.	SlaGenerator
Unnamed by Gyrd et al.	[16]	Risk-based threat analysis using risk scenarios, based on existing graph-based threat modeling method.	No tool
Unnamed by Haley et al.	[65]	Goal-based threat analysis, focused on system context and security satisfaction arguments. Results in security requirements	No tool
Unnamed by Haley et al.	[64]	Represent threat descriptions in tuples: action, asset, harm. Formal arguments are used to test the satisfcation of security requirements and reveal threats.	No tool
Unnamed by Huang et al.	[75]	Threat modeling on control systems and assessment based on the physical and economic consequences.	No tool
Unnamed by Kim et al.	[90]	Risk-based threat analysis, using use case diagrams and threat scenario templates to create reports.	Non-specific

Unnamed by Manzoor	[109]	Threat modeling for cloud ecosystems based on petri nets and design structure matrices.	No tool
Unnamed by Rhee et al.	[143]	Threat modeling of a mobile device management system by analyzing and identifying threat agents, assets, and adverse actions.	No tool
Unnamed by Tondel et al.	[171]	Combines MUC, attack trees, Security activity models.	Sea monster
Unnamed by Ware et al.	[191]	Actor profiles to derive threats, mapping threats to security objectives, and mapping objectives to security requirements.	Extension of Violet
Unnamed by Zografopoulos	[204]	Risk-oriented threat modeling specifically designed for cyber-physical systems.	No tool
VAST	[116]	Distinguishes between application and operation by mandating the inclusion of both application threat models (process flow diagrams) and operational threat models (architectural models).	ThreatModeler

4.6.2 Tools

As a result of the SLR twenty potential threat modeling tools have been identified. Note, that our focus was primarily on software tools, so tools such as the elevation of privilege card game have been disregarded. Table 4.3 provides an overview of the identified tools and the techniques or methods they support. These tools were included in our analysis only if they were explicitly mentioned in at least one of the papers. *"Corresponding paper"* denotes the paper sources that led to identification.

When examining each individual tool, we observed that the availability of threat modeling tools was very limited. Among the twenty identified tools, nine (45%) were not publicly available. Meaning, they could not be found with multiple Google searches or that the link provided in the paper led to a page that was not available anymore. Only two commercial tools were identified, the rest of the available tools provided a free download or were open source. We also observed that there was a variety of tool types. Seven tools (35%) could stand on their own. Meaning, they did not require any additional tools or concepts before they could be used. Besides visualizing the threat model, these tools were observed to actively assist in other activities of threat modeling. For example, the generation of threats. Certain tools were in the prototype, meaning the papers indicated certain future refinements and the absence of public availability. Furthermore, certain tools facilitated automated threat modeling, among which two were dedicated plug-ins designed for existing modeling software, while the remaining three were standalone quantitative tools. Among these tools, only SLAgenerator was available. However, the GitHub associated with this tool did not display active development.

The tools that have been identified will serve as input for the alternatives within the decision model. Due to the lack of availability and documentation, a substantial portion of these tools are

Table 4.3: The identified tools and the techniques or method they support. The availability denotes the public availability of the software. N/A denotes Not Available.

Tool Name	**Supports**	**Availability**	**Tool type**	**Doc.**	**Corres. paper**
ASTo	Apparatus Framework	Not available	Visualization tool	N/A	[114]
CORAS Web Tool	CORAS	Free online use	Visualization tool	[166]	[39, 105]
i* model tools	i*-based threat modeling	Free download	Visualization tool	[78]	[44, 102, 123, 124]
Microsoft TMT	STRIDE	Free download	Standalone tool	[119]	[82]
MUCSIM	Executable MUC	Not available	Plug-in for existing software	N/A	[194]
Octotrike	TRIKE	Open source	Standalone tool	[99]	[162]
OVVL	OVVL (framework)	Open source	Standalone tool	[153]	[151]
OWASP Threat dragon	STRIDE, LINDDUN, CIA	Open source	Standalone tool	[131]	[22]
P2CySeMol	Attack graphs	Not available	Quantitative TM tool	N/A	[74]
Plug-in for CASE	Automated analysis of security-design models	Not available	Plug-in for existing software	N/A	[82]
PRISM	IoTRiskAnalyzer	Open source	Model checker	[97]	[122]
Seamonster	Attack trees, Misuse case modeling	Open source	Standalone tool	[63]	[162]
Securi cad by foreseeti	Automated threat modeling	Not available	Quantitative TM tool	N/A	[42]
SecurITree	Attack trees	Commercial	Standalone tool	[101]	[146]
SLAgenerator	STRIDE	Open source	Quantitative TM tool	[59]	[60]
Tam2	Automated threat modeling	Not available	Prototype	[152]	[60]
ThreatModeler	VAST	Commercial	Standalone tool	[170]	[162]
Tiramisu	Attack paths	Not available	Prototype	N/A	[28]
UML4PF	Problem frames, MUC	Not available	Visualization tool	N/A	[9, 71]
Unnamed threat catalogue	Automated threat modeling	Not available	Knowledge base for automated threat modeling	[24]	[60]

not suitable for usage in a business context. Section 8.3.1 explains how this is dealt with to select and construct the alternatives.

4.7 Criteria for threat modeling selection

As shown in Figure 4.2 the SLR identified 32 papers that outlined certain criteria that can potentially be used to differentiate between threat modeling methods. It is found that the criteria used to compare threat modeling and other security-related methods can be qualities or features. Qualities are non-functional requirements such as ease of use and scalability. These are commonly assessed on a specific level such as high, medium, or low, these are further referred to as quality criteria. While features are aspects of the method that are distinctive characteristics or functionalities that can contribute towards improving a quality criterion. In this study criteria that contain a feature or a group of features, are referred to as requirements.

4.7.1 Requirements

Examining each paper, seventy potential requirements were identified. Due to the lack of studies that provide requirements specific to threat modeling and the lack of a common definition, these potential requirements were also extracted from related domains such as risk assessment and security requirement engineering. These requirements were not only used in literature to compare (components) of methods for selection but also for other purposes such as evaluation. During the extraction, it was qualitatively evaluated if these requirements could be applied in the context of threat modeling selection.

After the initial extraction, the unstructured requirements were re-evaluated and refined. A brainstorming session with multiple researchers that had expertise in threat modeling was conducted. This was deemed necessary to manage the significant complexity resulting from the integration of multiple intertwined domains with diverse terminologies. To illustrate, "the level of abstraction" by Shameli et al. [158] (threat modeling) is related to the "modeling view" by Tatam et al. [168] (risk assessment). Both have a category of criteria that contain a label for methods that focuses on assets. However, they also have other non-matching labels. Hence, a choice needed to be made on how to handle this situation, which lrompted the initiation of a brainstorming session. During this session the applicability to threat modeling was re-evaluated for each requirement, duplicates were removed, requirements that had a similar meaning were combined and the final set of requirements was split into eight different groups. This resulted in a collection of 41 different requirements, which is the basis for threat modeling method selection.

Table 4.4 displays the set of requirements extracted. *"Assessment"* denotes the values which are used to assess the requirement. This set contains requirements that can be assessed either with a Boolean value or a categorical value. As explained in Section 4.4, there are also certain criteria in the literature that are answered with a textual description. These are either combined into other requirements or disregarded due to them not being applicable for threat modeling method selection. The requirements are linked to a group based on their relevance to different aspects of threat modeling. For example, report representation is put into the output category, because it relates to how the output of threat modeling is displayed. Instead of doing a literature relevancy assessment, based on how often a requirement was mentioned in the literature, we used the complete set of requirements and validated the criteria based on expert input. Further details on the design of this validation can be found in Chapter 5.

TABLE 4.4: The requirements extracted from the SLR. Assessment denotes the values which are used to assess the requirement.

Requirement	Description	Assessment	Ref
Input			
Current security	The method takes current security practices into account.	Yes/No	[157]
Input type	The type of input that is required in order to start the threat analysis.	Attacker behaviour, Architectural design, Requirements	[157, 158]
Input data	Where the input data is originating from.	Historical data, expert opinion	[29]
Output			
Output type	The type of artifact that the method produces.	Security requirements, threats, mitigations	[107, 140, 158]
Output granularity	The level of detail in which the result from threat modeling is presented.	High-level: descriptions of threats, low-level: source code	[175]
Report representation	How the findings from threat modeling are displayed in the report.	Structured text, model-based	[22, 175]
Mitigation strategy	Whether or not mitigation strategies are provided by the method.	Yes/No	[159]
Perspective			
Approach	The central ideology that the threat modeling method is based on.	Attack-centric, risk-centric, Security requirement engineering,	[175]
Focus	The viewpoint from which the technique is performed.	Attacker, Defender	[159]
Agents	Does a threat model focus on one perspective or multiple perspectives at the same time.	Multi-agent or Single-agent	[49, 124]
Modeling view	The level of detail on which the actual model is created.	Architecture, behavior or platform	[57]
Modeling notation			
Abstraction	The level of detail at which the components are represented in the notation.	Goal-based, Model-based, problem-based, process based	[22]
Formality	Distinguishes between formal and graphical representations.	Formal, Graphical	[175]
Language	Considers the notation that the technique uses in order to model the threats	E.g. DFD, Petrinets	[22, 50, 57, 107]
Similarity with software specification languages	The notation used is similar to already known software specification languages.	Yes/No	[34]
Extensibility	The ability to model domain specific components.	Yes/No	[22]
Coverage of components	Regards the presence of the most common threat modeling components.	Yes/ No (For each component)	[34]
Relationships between components	Regards the presence of the most common possible relationships between components.	Yes / No (For each relationship)	[45]
Countermeasure impact	Determines if the threat model can show the impact of a countermeasure on vulnerabilities and attacks.	Yes/No/Partial	[45]
Modeling context			
Barriers for practitioners	Considers if there are any knowledge barriers for executing the threat modeling method.	Yes/No	[157]
Layer	The level of system details at which threats are assessed.	Environment, total system, computer-based system or software	[140]
Applicability to system	The method is designed for either general use or a specific type of computer-based system.	Generic, specific	[50, 140, 159, 200]
Threats to hardware	Looks at if threats to hardware such as physical tampering are taken into account.	Yes/No	[157, 159]
Method process			
Threat generation	How the process of identifying and establishing a list of potential threats is done.	Automatic or manual	[22, 168, 200]
Involved entities	Considers the recommended parties that should be involved (e.g. management, business-line).	E.g. Security specialists	[157]
Set a boundary	The technique explains how to define the limits of the system, so that we can identify the area that needs protection.	Yes/No	[157]
Threat analysis			
Resource valuation	The analysis differentiates between non-critical and critical resources.	Non-critical, Critical	[157, 158]
Effect propagation	The analysis shows how attacks on one part of the system can affect other parts that depend on it.	Propagated, Non-propagated	[45]
Quality assurance	Assesses if quality assessment is a part of the analysis procedure.	Explicit, present, not present	[49, 130, 175]
Threat library	A list of known threats (a threat library) is used to identify applicable security threats based on the system's components.	Yes/No	[28, 175, 201]
Analysis level	The level of detail used to analyze the system for potential threats.	Architecture, design and source code	[3]
Security objective	The areas of security covered by the method.	Confidentiality, integrity, availability	[49, 50, 107, 130, 175]
Prioritization	Includes a method for prioritizing threats.	Yes/No/Partial	[45]
Risk	The role of risk in the threat modeling method.	Not considered, Internal part of technique	[175]
Steps of vulnerability exploitation	Time or steps to exploitation is calculated as a part of the method.	Yes/No/Partial	[45]
Threat analysis type	Differentiates between quantitative and qualitative techniques for threat assessment.	Qualitative, Quantitative	[28, 29, 200]
Other			
Stopping condition	The method has a definition of done.	Present, not present	[175]
Interoperatibility	Assesses if the technique interact with other security techniques.	E.g. Yes, is part of risk assessment	[140]
Validation	Validation or verification of model, interaction and conflicts of components are shown.	Yes/No	[49, 130]
Automation	Considers if the tools have automated solutions.	Yes/No/Partial	[159]
Portability	The ability to transfer models from one machine or system to another.	Yes/No	[159]

4.7.2 Quality criteria

Besides a set of requirements, seventeen individual quality criteria and ten quality groups (containing multiple individual qualities) were identified. Like the requirements, the qualities were extracted from comparisons beyond the threat modeling domain. Moreover, they were evaluated based on their suitability in the context of threat modeling selection. The quality groups were split up into individual qualities and the set was refined by combining qualities and removing duplicates. Afterward, they were subjected to a brainstorming session with multiple researchers, in which the applicability was reassessed, and quality criteria were reformulated and structured. After these refinements, a total of 27 different qualities were found suitable in the context of threat modeling.

A hierarchy was designed to structure the quality criteria. This was achieved by adapting the internal method quality criteria by Brinkkemper et al. [19] and utilizing those criteria as groups. Furthermore, these groups were populated with qualities found in the SLR. A total of 15 qualities were directly assigned underneath these quality groups. Additionally, 12 criteria were assigned as quality sub-criteria and were placed underneath a quality criterion. Assigning the qualities was done based on their hypothesized contribution. For example, it was hypothesized that completeness ("Does the method specify a state or condition of having the necessary or appropriate parts of a threat model.") contributes towards a semantically correct and meaningful method (reliability).

In Table 4.5 the quality hierarchy is presented. Note that several definitions have been reformulated to fit the context of threat modeling method selection. When examining the corresponding papers, we found that several distinct assessment scales are used to benchmark these qualities. For example, Raspotnig et al. [140] did a quality assessment given a textual description. While others used a scale with three levels (high, medium, and low) [90, 159]. Occasionally scales with both three and five levels were used interchangeably [111]. To avoid conflicts between the assessment scales we decided that each quality presented in this hierarchy can be assessed with low, medium, or high.

TABLE 4.5: The quality hierarchy created as a result of conducting the SLR. Note that some definitions have been reformulated to fit the context of threat modeling method selection. All qualities can be assessed on a three-level scale (high, medium, or low).

Criteria group	**Quality criteria**	**Quality sub-criteria**	**Description**	**Ref**
Efficiency			The method can be performed at minimal cost and effort	[19]
	Cost		The amount of resources to produce a threat model. Both the cost for creating a threat model (time/effort) and the cost for the purchase of a threat modeling tool are considered.	[159]
	Reusability		The ability to use parts of one threat model to create a new model.	[34, 43]
	Tailorability		Capability of a method to adapt to a specific application.	[156]
	Scalability		If a method is applicable to large systems or not. There are components in the method that scale with the size and complexity of the system under assessment.	[140, 159]
Reliability			The method is semantically correct and meaningful	[19]
	Completeness		Does the method specify a state or condition of having the necessary or appropriate parts of a threat model.	[34, 89, 90, 110, 111, 117, 150]
	Precision (Ambiguity)		How consistent the results of threat modeling are when conditions are the same.	[90]
	Documentation		The documentation of the method is clear and extensive.	[156, 159]
		Technique documentation	The technique is documented clearly and extensively.	[156, 159]
		Tool documentation	The tool has a clear and extensive documentation.	[156, 159]
	Software evolution support		How well a threat modeling method supports the continuous change and improvement of software throughout its lifecycle.	[34]
		Modularity	How easy is it to develop and use software components separately.	[34]
		Component architecture	The level of support for a component-based structure that allows software modules to be added and removed with ease.	[34]
		Change propagation	Ability to keep track of changes made to system and to guarantee that a change is correctly propagated such that no inconsistent dependency is left unresolved.	[34]
		Change impact analysis	Ability to evaluate the effect that changes made to specific artifacts will have on other system components.	[34]
	Suitability		The aim of the method is in line with the expectations of the stakeholders.	[29, 156]
	Maturity		How well the different parts of the method are structured and designed.	[110, 156, 159]
		Technique maturity	The technique supported by a systematic and structured process.	[110, 140, 156, 159]
		Tool maturity	Degree of optimization of the tool for standard usage. (Values based on the capability maturity model)	[110, 156, 159]
	Support for maintainability		How easy it is to make and track changes while using the method.	[89]
		Modifiable	The ability to make changes to an existing threat model.	[89]
		Traceability	Concerns the ability to trace the history, application and location of threat modeling components.	[34, 89, 117]
	Understandability		How easy is it to understand the outcome of the method.	[34, 90, 117]
Applicability			The users are able to apply the method	[19]
	User experience		The experience of a person who uses a threat modeling method.	[22]
		Ease of use	Simplicity of use of the technique, tool, and notation.	[89, 90, 110, 111, 150, 156, 159]
		Learning curve	The cost associated with learning the threat modeling technique, tool, and notation.	[156, 159]
	Compatibility with agile development process		The technique can be used within an agile environment.	[159]
	Compatibility with related processes		How well the threat modeling method works with other (security) related processes such as risk assessment. (e.g. How much overlap is there between the processes?)	[189]

5 Developing data collection tools

This chapter describes the activities that were performed in preparation for the expert survey and interviews.

5.1 Content refinement

Before data collection was initialized, the interview as described in Section 3.2.2 was tested in a pilot interview with a security expert. In addition to rectifying minor spelling errors, the pilot interview led to significant alterations in the interview's guide and its structure. These adjustments were made based on the insights gained from the pilot interview, that aimed to enhance the overall quality of the interview process. We describe the most important modifications in the remainder of this chapter.

5.1.1 Introducing user stories

One of the takeaways from the pilot interview was that the participant did not understand how to answer the questions related to the literature validation (Part 4). This issue arose due to the terminology gap between literature and practice, but also because the template provided to the participant contained multiple tasks. This complexity not only hindered comprehension of what aspects to reflect upon but also posed challenges in formulating suitable responses. Therefore, it was decided to reformulate the requirements in the format of user stories.

When a requirement had a binary assessment (yes/no), one user story was created to capture the requirement. When a requirement could be answered with a textual category, each element was rewritten in a separate sub-user story to ensure the modularity of each component. The original description was either retained or slightly modified in the case of an overarching (main) requirement, to preserve the group-like structure. Additionally, the group to which the requirements are assigned was rewritten into an epic by providing it with an elaborate description (See Table 5.1). Furthermore, standard definitions were added for terms that were not familiar to the partitioner (See Table 5.2). Moreover, examples were added where necessary. This was minimized because we observed that an example often only describes one angle of the requirement or quality, which could lead to a bias when reflected on. The transformed requirements from the literature can be found in Appendix B.2.

5.1.2 Splitting up participants

It became apparent that asking practitioners to assess the connections between specific qualities and the necessary requirements involved too much abstract thinking. Consequently, two distinct types of interviews were introduced.

1. Requirement validation & exploration interviews
2. Dedicated relationship interviews

During the requirement validation and exploration interviews, participants were asked both exploratory questions and questions that evaluated requirements and quality criteria found in the literature. The purpose of these interviews was to assess the literature's findings and incorporate requirements and qualities from the organisations' perspectives. Additionally, the interviews involved discussing the current approach for selecting a method, the perception of what threat modeling is, and the significance of tools and techniques when conducting threat modeling. Through

TABLE 5.1: The requirement groups from the SLR written as epics.

ID	Epic name	Description
E001	Input	Everything that happens before starting the threat analysis. This regards components that are related to identifying and describing what is already in place in the organisation.
E002	Output	Requirements related to the outcome of the threat modeling method.
E003	Perspective	Requirements related to the perspective of the threat modeling method. Contains components that represent the point of view from which the approach solves the threat modeling case
E004	Modeling notation	Requirements specific to the notation used to represent the threat model.
E005	Modeling context	These requirements are about the context in which the modeling method can or cannot be applied, which is called the scope of usage.
E006	Method process	Requirements that apply to the process of threat modeling and its environment.
E007	Analysis	Requirements that are related to the threat analysis. This mostly regards the steps involved, the reasoning used, and the aspects to be considered during the analysis."
E008	Other	Requirements that are not placed in a category, explicitly relate to the integration with the SDLC or relate to tool support.

these interviews, it was possible to identify relationships between qualities and requirements by examining how a participant discussed a specific requirement within a quality context (e.g., "It is easy to use when...").

As a way to systematically examine the relationships between components and evaluate hypothesized relationships, dedicated relationship interviews were done. These interviews were conducted with participants who are currently working as researchers or have previous experience in threat modeling research. Participants were asked to read through the qualities and requirements before the interviews to familiarize themselves with the content. During the interviews, the qualities were presented and discussed one by one, and participants were asked to brainstorm about elements that could be incorporated into a threat modeling method to positively contribute to these qualities.

Table 5.2: Definitions utilized within the user stories that were not familiar to the practitioner.

Ref	Term	Definition
[A]	Attacker behavior	A list of expected actions and tactics used by a malicious actor to exploit vulnerabilities in computer systems.
[B]	Security assumptions	Underlying beliefs about the operational design of a system that can influence its security posture. In other words: Ideas about how a system should work that can affect how secure it is.
[C]	System goals	Intended outcomes or objectives that the system under analysis is designed to achieve. These goals can be documented in a high-level description.
[D]	Functional behavior	Functional behavior of the system refers to the intended or expected actions and outputs of the system. For example, a payment system transfers a balance from one account to the other, using the internet. When there is a DDOS attack on the system, this threat results in the function being denied. Meaning that the system is not able to transfer balance.
[E]	(Secuirty) Goal	Intended (security) outcomes or objectives that the system under analysis is designed to achieve. An example of a security goal can be: To ensure that only authorized users are able to access the information (confidentiality).
[F]	(Secuirty) problem	Vulnerability in a system that can be exploited by attackers. For example, using TLS 1.0 in your system is a security problem.
[G]	Knowledge barriers	Knowledge barriers are obstacles or limitations that prevent practitioners from being able to perform threat modeling. For example, a cursus is required to perform a specific threat modeling method.
[H]	Steps to vulnerability exploitation	Refers to the steps an attacker may take to identify and exploit a vulnerability. For this research we measure this in term of estimated time until a vulnerability is exploited.
[I]	Qualitative	Qualitative threat modeling methods are more subjective, often based on brainstorming, expert opinion and non quantifiable techniques. It does not require much data and details but rather relies on diagrams and checklists.
[J]	Quantitative	Quantitative threat modeling is based on data and statistical analysis in order to quantify the effect of security threats. This often involves mathematical models or simulations of different types of attacks.

Table 5.3: Demographics of the domain experts. "N/A" indicates that a domain expert did not participate or the interview did not sufficiently cover the preset topics.

ID	Document ID	Residence	Current role(s)	Highest degree	Experience	Int. type
P1	Interview 1	NL	Infomation Security Officer	Master's degree (University)	0-5	1
P2	Interview 3	UK	Application Security lead	Bachelor's degree (University)	25-30	1
P3	N/A	USA	Chief Security Architect and CTO	Master's degree (University)	35+	1
P4	Interview 4	NL	Infomation Security Officer	IS Certification	30-35	1
P5	Interview 5	NL	Technical Product Owner	Master's degree (University)	20-25	1
P6	Interview 6	NL	Privacy and Security engineer	Bachelor's degree (University)	20-25	1
P7	Interview 7	NL	Operational Security manager	Applied sciences bachelor (HBO)	25-30	1
P8	Interview 14	BE	Postdoctoral researcher	PhD	5-10	2
P9	Interview 16	USA	Professional in residence and CTO	Master's degree (University)	35-40	2
P10	Interview 15	Spain	Professor	Master's degree (University)	25-30	2
P11	Interview 8	NL	Infomation Security Officer	Undisclosed	25-30	1
P12	Interview 9	BE	Infomation Security Officer	Bachelor's degree (University)	10-15	1
P13	Interview 2, 13	UK	Security director	Master's degree (University)	5-10	1, 2
P14	N/A	UK	Security consultant	Bachelor's degree (University)	15-20	1
P15	Interview 10	USA	Application security engineer	Post-master	15-20	1
P16	Interview 11	USA	Security engineer and consultant	Bachelor's degree (University)	20-25	1
P17	Interview 12	Sweden	Security consultant	Master's degree (University)	5-10	1

Although this is still open-ended, by creating a modular structure as well as simplifying the task, the participants were able to complete the task. Specifically, these interviews helped to identify relationships for less commonly mentioned qualities, such as scalability and modifiability.

Table 5.3 shows the country of residence, the current role, the highest obtained degree, and years of experience of the experts. The *"Document ID"* refers to the transcript(s) of the interview(s) in which the expert has participated. "N/A" is assigned when candidates either did not want to partake in the interview after the survey was completed or the actual conversation did not adequately cover the pre-defined topics. *"Interview type 1"* signifies the participation in the requirement validation and exploration interviews, whereas *"type 2"* denotes participation in the dedicated relationship interviews.

5.1.3 Pre-interview survey

One key lesson learned from the pilot was the importance of time management. The pilot interview revealed that the initial set of tasks took three times longer than initially predicted. Consequently, significant modifications were made to the interview's structure to address this issue. The participants of the validation and exploration interview were requested to fill in a survey beforehand. This survey required participants to answer a set of questions from the perspective of a decision-maker responsible for selecting a threat modeling method for a development project. On average, participants spent approximately 45 minutes to complete the survey, which was divided into three main parts.

The first part comprised a single open question asking the participant to describe how they would select a threat modeling method for their organisation. Throughout the survey, the definitions for a method, technique, tool, and notation were provided at the top of each section, aiming to assist practitioners in distinguishing between these elements effectively. The subsequent part contained questions regarding the quality criteria for selecting a threat modeling method. Participants were asked to rank the qualities relative to the other criteria of the group they belonged to, thereby determining their relative importance and potentially resolving ties in the final decision model. Moreover, the participant had the option to remove, modify or add criteria within the quality hierarchy.

The final section of the survey encompassed questions pertaining to the presented requirements in Appendix B.2. This part of the survey was divided into eight distinct sections, each presenting requirements one by one, categorized based on their corresponding epic. Within this section, two types of questions were posed. The first type measured the relevancy of the main requirements (identified with IDs like US101), for selecting a threat modeling method. This required a simple yes or no (relevant / not-relevant) answer. The option to select *"no opinion"* was provided for cases where the requirement was not fully understood. The second set of questions pertained to the sub-user stories (IDs like US102a). In that instance, the participant was presented with a multiple-choice question that asked to select all options that are applicable given their experience. Additionally, participants had the option to suggest other user stories or indicate if none of the provided options were relevant. Throughout the survey, the participants were encouraged to leave comments when anything was unclear or should be modified. The complete survey can be found in Appendix F. These comments, along with individual survey responses, were reviewed by the interviewer in preparation for the interviews. Key topics of conversation were identified and highlighted, facilitating discussion and structuring the interview process.

6 Pre-interview survey results

This section contains the findings of the pre-interview survey. A total of fourteen surveys were completed. Participants were hand-selected according to the guidelines mentioned in Section 3.2. Although the number of participants is limited, purposive sampling focused on threat modeling experts facilitated the validation of the diverse set of complex terminology extracted from the literature. Moreover, reviewing each individual response yielded insight into their perspectives, which was then used as a basis for the in-depth interviews. The demographics of the expert participants can be found in Section 5.1.2. The full survey results are available in Appendix B.

6.1 Requirement survey validation

To contribute towards answering the second sub-research question: *What are the criteria for selecting threat modeling approaches?*, the requirements obtained from the SLR were presented in the third part of the survey. The participants evaluated how relevant the main requirements were and also assessed the applicability of the sub-user stories. 42 main requirements were tested and on average our participants marked 77.7% as relevant. Also, 51 sub-user stories were questioned. The average practical applicability of the sub-user stories was 62,4%. All results from the requirement validation part of the survey can be found in Appendix B.2. The upcoming sections will mention the highlights of the results.

6.2 Relevancy of main requirements

There were no clear outliers within the main requirements. The lowest rated requirement was US603 - set boundary: *"As a decision-maker, I want the technique to clearly explain how to define the limits of the system, so that we can identify the area that needs protection."* Which was rated as relevant by 50% of the participants. This indicates that the requirements thus far are already highly relevant. The next lowest are US504, US709, US710, and US804 with a relevancy rating of 57,1%. US504 is about considering hardware threats and US709 refers to the requirement of considering steps to vulnerability exploitation. US710 is differentiating between qualitative and quantitative approaches and US804 is about verification support in a tool. The survey data did not provide a clear explanation of why these were ranked at the bottom.

There were three requirements that were rated relevant by all participants (US304, US502, and US706). All of these pertain to the details of the threat modeling method. More specifically, US304 is about the level of detail in which the actual system is represented. For example, the system can be represented as a set of software components or behavioral rules. US502 and US706 are both focused on the level of detail at which threats are identified. Where US502 focuses on the level of the system and US706 relates to the coverage of threats in correspondence with the CIA triangle. Based on these results, it can be inferred that the decision-maker considers the level of detail at which threat modeling is performed to be highly relevant. An overview of the least and most relevant main requirements can be found in Table 6.1.

6.3 Applicability of sub-user stories

The average applicability of sub-user stories was noticeably lower than the relevancy of the main requirements. Several participants commented on the applicability questions, highlighting that it was occasionally driven by personal preference rather than general applicability. This was taken into account during the requirement refinements for the decision model. Any sub-user stories that are considered applicable by less than 50% of the participants are shown in Table 6.2. These are now shortly discussed.

The lowest applicable sub-user stories, US402a and US710b, are closely related. It could be suggested that having a formal notation is advantageous when conducting quantitative threat modeling. However, based on the comments in the survey, using quantitative threat modeling is currently not applicable. One of the reasons provided by participants is that although threat modeling based on a mathematical model would be desirable, it is not yet feasible due to the associated high cost. On the other hand, all participants found that using a graphical notation was practical. This suggests that there is a significant difference in the applicability of graphical and formal notations.

Table 6.1: The least and most relevant main requirements. R stands for Relevant, NR denotes Non-Relevant and NO means No-Opinion. The percentage indicates the column contains the cumulative percentage. The relevancy of all requirements can be found in Appendix B.2.

ID	Requirement	User story - Description	R	R%	NR	NR%	NO	NO%
US603	Set boundary	As a decision-maker, I want the technique to clearly explain how to define the limits of the system, so that we can identify the area that needs protection.	7	50%	5	35.7%	2	14.3%
US504	Hardware threats	As a decision-maker, I want to have a threat modeling method that considers threats to hardware.	8	57.1%	4	28.6%	2	14.3%
US709	Steps of vulnerability exploitation	As a decision-maker, I want to have a threat modeling method that considers steps to vulnerability exploitation[H] (expressed in time) as part of the analysis.	8	57.1%	5	35,7%	1	7.1%
US710	Threat analysis type	The differentiation between quantitative and qualitative techniques for threat analysis.	8	57.1%	4	28,6%	2	14.3%
US804	Verification	As a decision-maker, I want to receive verification support for the threat model, so I can guarantee the model is built correctly.	8	57.1%	3	21.4%	3	21.4%
...	...	...	...	...	...	...	...	...
US304	Modeling view	The level of detail on which the actual model is created.	14	100%	0	0%	0	0%
US502	Layer	The level of system details at which threats are assessed.	14	100%	0	0%	0	0%
US706	Security objective	The areas of security covered by the method.	14	100%	0	0%	0	0%

US705a and U102a revolve around the role of source code in threat modeling. The survey results clearly indicated that the use of source code in threat modeling is not applicable in practice. This observation is best exemplified by the following statement written by one of the participants: *"I would rather use SAST and DAST tools than threat modeling at the point source code has been built."* [P13] Essentially, there are alternative techniques and tools that become more relevant once the source code is built. This aligns with the literature, which suggests that threat modeling is best applied in the early stages of the development life cycle [185].

Among the participants, 35,7% found US301d applicable while 28,6% found US301e applicable. These are both related to requirement engineering. Although they were not determined to be the least applicable, certain participants noted that they were unfamiliar with these approaches and marked them as non-applicable accordingly. A similar explanation was given for US601b, which regards threat generation based on expert knowledge. This sub-user story was not found applicable by several participants because they were not familiar with the example method provided. It is noteworthy that only 35,7% of the participants found this sub-user story applicable, although the survey results showed a preference towards more qualitative methods that often involve experts.

Table 6.2: The least applicable sub-user stories by domain experts. A denotes Applicability. The percentage indicates that the values in the column display the cumulative percentage. The applicability of all sub-user stories can be found in Appendix B.2.

ID	**User story - Description**	**A**	**A%**
US402a	As a decision-maker, I want the notation to be formal, so that threat modeling based on a mathematical model is possible.	1	7.14%
US710b	As a decision-maker, I want my threat modeling method to be quantitative, so it is more detailed.	1	7,14%
US303b	As a decision-maker, I want one model to show the perspective of a single agent, so that the model does not become too complicated. (For attack-centric approaches)	3	21.43%
US705a	As a decision-maker, I want the depth of the analysis to reach source code level.	3	21,43%
US708c	As a decision-maker, I want to have a threat modeling technique that does not consider risk. (E.g. because it is not needed to obtain the prefered threat modeling outcome)	3	21,43%
US102e	As a decision-maker, I want the analysis to be done based-on source code, so that the analysis is performed based on the actual system.	4	28.57%
US301e	As a decision-maker, I want the technique to be centered around Security Requirements Engineering (SRE), so that the primary goal is to identify security requirements.	4	28.57%
US401d	As a decision-maker, I want the model notation to represent its components centered around the business processes.	4	28.57%
US301d	As a decision-maker, I want the technique to be centered around Goal-Oriented Requirement Engineering(GORE), so that both functional and non-functional (among other, security) requirements are obtained.	5	35.71%
US401a	As a decision-maker, I want the model notation to represent its components centered around (security) goals.	5	35.71%
US401c	As a decision-maker, I want the model notation to represent its components centered around (security) problems.	5	35.71%
US601b	As a decision-maker, I want to have a threat modeling method that generates threats based on expert knowledge. (e.g. CORAS)	5	35.71%
US103b	As a decision-maker, I want the threat assessment to be done based-on historical data.	6	42.86%
US203a	As a decision-maker, I want the report from a threat modeling project to-be a structured text.	6	42.86%
US301a	As a decision-maker, I want the technique to be attack-centric, so there will be a focus on identifying attacker profiles and the complexity of attacks.	6	42.86%

Less than 50% of the participants found the use of historical data (US103b) to be applicable. A participant suggested that this might be due to companies not commonly collecting this type of data. Additionally, it was strongly suggested that historical data alone is insufficient for threat identification.

US401a, US401c, and US401d all have to do with the level of detail in which the components of the model are represented. These sub-user stories demonstrate relatively lower applicability compared to US401b (57,1%), which states: *"As a decision-maker, I want the model notation to represent its components centered around a model of the system."* The difference in applicability indicates a preference for visualizing the system as part of a threat modeling method. Moreover, it is strongly preferred to have multiple agents as part of a threat model (US303a) over a single agent (US303b) with an applicability of 78,6% and 21,4% respectively. Lastly, US708c is only applicable for 21.4% of the participants, while considering risk as part of the method has a 78,6% applicability and considering risk externally has a 50% applicability rate. This may indicate that risk is a desirable component of threat modeling in practice.

6.4 Additional requirements

Several additional requirements were mentioned in the comments of the survey and through the "other" answer option. Most of the requirements were for a tool, which was not well-represented in the results of the SLR. For example, *"If we are talking about tools, I want: - versioning - secure storage of the model - RBAC for the model."* [P2] We translated these comments into the following requirements:

- **Custom priority:** As a decision-maker, I want the ability to replace the existing threat prioritization in the tool with a priority that is specific to our organisation.
- **Layer decomposition:** As a decision-maker, I want to have a threat modeling method that allows switching between system layers (C4 model as reference).
- **Versioning:** As a decision-maker, I want the tool to have the capability to maintain a record of the versions of threat models, allowing me to track the changes made over time.
- **RBAC:** As a decision-maker, I want access to the threat models to be restricted by RBAC so that only the appropriate stakeholders can view and edit them.
- **Secure storage:** As a decision-maker, I want the tool to store the threat models in secure storage, so confidentiality is insured.

- **Classes and instances:** As a decision-maker, I want the tool to distinguish between classes and instances of components, so customized threat assessments based on each specific instance can be done.

Furthermore, several sub-user stories were incorporated based on the additional input received from survey participants. For example, when asked about input type one participant wrote: *"Personas usually work best and these are often malicious. ... the majority of the work is done with evil personas."* [P6] We translated this comment into the sub-user story *"Input type - Personas"*. All additional requirements were subjected to a more detailed examination during the interviews.

- **Input type - Business processes:** As a decision-maker, I want the analysis to be done based on business processes so that the user knows the context.
- **Input type - Personas:** As a decision-maker, I want the threat assessment to be done based on personas.
- **Modeling view – Process:** As a decision-maker, I want the threat model to describe the process surrounding the system.
- **Security objective – Privacy:** As a decision-maker, I want the security objective of the threat modeling method to cover privacy.

6.5 Quality criteria survey validation

In the second part of the survey, the quality criteria extracted from the literature were queried. Instead of asking about their relevance directly, participants were shown the hierarchy of quality criteria and asked to rank them within their respective groups. The following remark was placed to grasp the relevancy: *"If you come across any quality criteria that is not important at all, you can put them in last place and mention this at the bottom of the page."* This was followed up by questions Q1.6 and Q1.7, in which participants were asked to modify or remove any of the quality criteria. Although several comments were made, none of the participants proposed a direct change or removal of a criterion. However, their opinions and struggles were noted and used as input for individual discussions during interviews, which helped refine the quality criteria.

Based on the answers to these questions, a couple quality criteria were added. These criteria were also explored during the interviews:

- **Grokkiness:** The intuitiveness of the threat modeling methods for the stakeholders intended to apply it.
- **Testability:** To what degree the recommendations that result from the threat modeling exercise are applied and re-evaluated.

6.6 Additional findings: Relative quality importance

All participants ranked the quality criteria within their hierarchical group. Originally, this data was collected to facilitate the creation of a tiebreaker mechanism in case the alternatives from the decision model obtained an equal score. In such a scenario the inference engine could solve the tie, for example, by summing the relative importance of all the requirements marked with a *Must have* for the tied alternatives. Due to resource constraints, this study chose not to consider relative weights in the final decision model and decided to leave this opportunity for future research. The full relative ranking per category can be found in Appendix B.1. Nevertheless, key takeaways will be briefly discussed.

According to the domain experts that took the survey, applicability is the most important quality category for a threat modeling method. The ability of users to apply the method is considered more important than efficiency and reliability. In the quality category of applicability, the user experience was ranked highest, while the compatibility with related processes was most frequently ranked lowest. This suggests that the practitioners place significant importance on the user when choosing a threat modeling method. This was further confirmed by the fact that half of the participants chose understandability (*"How easy it is to understand the outcome of the method as their first choice."*) as their top choice in the reliability category. Both of these qualities are related to how people perceive the interaction with the method.

When examining the sub-criteria that distinguish between a tool and a technique's quality, it became apparent that the participants had a clear preference for qualities related to techniques. Specifically, technique documentation (85,7% ranked #1) and technique maturity (78,6% ranked #1) were often valued over their tool variants. This suggests that there may be a discrepancy in the significance of different components within a method.

Other findings show that, according to the domain experts, being modifiable is more important than traceability (71,4% vs. 28,6% ranked #1). Additionally, within the user experience quality, there was a strict tie between the importance of ease of use and learning curve sub-qualities. While these results are not directly used as part of the decision model, they provide valuable insights into the context of what practitioners consider important when choosing a threat modeling method and can guide future research in this area.

7 Expert interviews

This chapter presents the results of the expert interviews, which were conducted in two different formats.

7.1 Requirement validation interviews

A total of twelve interviews of this nature were successfully conducted with domain experts who also completed the survey. The demographics of these participants are described in Section 5.1.2. Specific participants are further referenced using their assigned ID. The interview itself was divided into two equally sized parts.

In the first part of the interview, the current selection of threat modeling methods was explored. This included a detailed discussion based on the answers provided in the survey. The conversation then shifted toward discussing threat modeling tools and exploring their significance in the decision-making process. Additionally, participants were asked about the organisational aspects related to selecting a tool. Throughout this conversation, various tool features were mentioned, often related to existing solutions. Therefore, participants were encouraged to consider additional requirements that would be important if they were not limited by existing solutions.

Subsequently, the role of a technique in threat modeling was discussed, including its importance and general factors that should be considered when selecting a technique. This part of the conversation offered valuable insights into the selection problem from the perspective of practitioners, which helped refine the decision-making method and identify additional requirements and quality criteria from an organisational standpoint.

During the next part, the participant's survey answers were reviewed. The interviewer highlighted certain answers for further discussion, specifically asking for more information on particular selections made. The highlights primarily aimed to discuss and clarify the least relevant requirements mentioned in the survey. Additionally, they served to initiate discussion on the following aspects:

- Potential inconsistencies in the survey answers.
- Survey questions that were answered with *"no-opinion"*.
- Outliers that were marked as not-relevant or not-applicable, even though they are generally accepted by the sample.
- Contributing factors to the highest-rated qualities (For matching qualities and requirements).

Discussing the highlights assisted in gaining an understanding of how practitioners interpret the requirements and which concepts they use to describe them. This served as input for refining the set of requirements to align with the terminology commonly used in practice. The changes made based on these observations can be found in Appendix D.2.

7.1.1 Qualitative coding

For the analysis of the interviews, a multi-phase approach was employed for the qualitative coding of the transcripts. Several interviews were conducted in Dutch. Before starting the coding process, the Dutch transcripts were translated into English. They were translated while keeping in mind the goal of accurately representing the content and essence of the interviews. Following the guidelines of Saldana [148] on how to do qualitative coding, a 4-phase process was designed and executed.

Phase 1: Initial coding. The first phase is initial coding in which the researcher thoroughly read through each transcript and marked segments that could potentially contribute towards answering SRQ1 or SRQ2. Furthermore, statements and examples that captured the thought pattern of the domain expert were highlighted. During this phase, a wide variety of information was captured, and off-topic segments were filtered out.

Phase 2: Creating a codebook. Upon completion of phase 1 a codebook was developed. The previous research steps enabled the use of the grouped user stories and the quality criteria hierarchy as a foundation. As the coding process unfolded, additional qualities and requirements were identified and integrated into the existing structure. Moreover, two categories of codes were added. The first category, labeled "Current selection", encompassed codes that described tasks and preferences related to the selection of threat modeling methods. The second category, named “Other”, contained observations that did not directly contribute to answering the research questions, but provided valuable context for the research. This ranges from the perceived goal of threat modeling and its strengths and weaknesses to codes that highlight inherent beliefs, assumptions, or definition misalignments.

Phase 3: Axial coding. This phase involved refining the codebook created in phase 2. Codes were combined and renamed for clarity, duplicate codes were dealt with and irrelevant codes were removed. To ensure thoroughness, the codebook was reviewed multiple times, and cross-coding by another researcher was conducted to support the final refinements.

Phase 4: Selective coding. In the selective coding phase, the primary objective was to develop codes related to requirements and qualities. As interviews included specific questions about highlighted criteria, codes were only assigned when they were naturally brought up by the interviewee. An exception was made for assigning quality codes in the segment that specifically inquired about requirements contributing to the highest-rated qualities from the participant’s survey. This was employed to capture potential relationships between the most important qualities and requirements. Additionally, codes related to the current selection and threat modeling goal were applied and further developed.

After completing the coding process, the coded segments were analyzed using a combination of manual and automatic methods. This included automatic occurrence analyses and the qualitative interpretation of coded segments. The results of the analyses are reported in the following sections.

7.2 Current selection approach

It was observed that there was a conceptual difference among participants when asked how they would currently choose a method. Certain participants explained their selection in the context of a tool while others focused on a technique: *"I would start at looking at which tools exist on the Internet ..."* [P1] versus *"If you look at the different models, ... I particularly look at ... and then you can still say: PASTA is nice, but ... to be able to look at it more from the technique point of view."* [P11] This has to do with the previously discussed paradigm.

The experts that focused on a tool, explained that they would approach the selection similarly to how they would select other security tools. They would initiate the process by conducting a requirement analysis, where the objective they intend to achieve through threat modeling can potentially play a crucial role in prioritizing these requirements. Subsequently, market research is conducted and the identified tools are evaluated based on the features they offer. Then the tools are compared in terms of requirements satisfaction. Following this, a qualitative or instinctive decision is made to select the most suitable tool.

When the selection was not centered around a tool, the method was often selected based on attributes related to the organisational context. This was closely associated with the organisational fit. Specifically, the majority of the experts considered the security maturity of the stakeholders that are involved in the threat modeling activity. When asked to elaborate, a participant stated: *"So specifically, this is about the security maturity. How aware people are of the need for security and frankly, the higher that is, the more effortful a model I'd be willing to go for because they often give better results and they give clearer, more standardized results."* [P13] This was interchangeably addressed as the security maturity of their organisation. In cases where the security maturity is low, practitioners recommended a lightweight approach, often utilizing a predefined list of topics or questions to identify threats. For organisations with a high security maturity, a more elaborate and structured format was suggested, where additional threat generation through user input and tool support become more important. Note, regardless of the security maturity, the human aspect of threat generation remained a crucial part of threat modeling.

Regardless of their view on threat modeling or the security maturity of the organisation, the adoptability or the intention to use by the team emerged as a central thought behind the selection of multiple domain experts. One participant explained this by stating: *"I find that in most cases the problem is not because organisations are doing threat modeling, but they're not following the perfect method. The biggest problem is that they do not do it at all."* [P16] Various selection approaches explained by the experts revolved around the user aspect. For example, certain experts selected a technique or tool that matches the current way of working or selected a method based on the technical capabilities of the user. Certain experts even selected a tool directly suggested by a user to ensure its adoption: *"It should come from the people who are going to use the tool, not from people who are going to dictate the use of the tool. ... They know what they want. ... And they find a tool that they want, so it eliminates the need for all of the evaluations. ... That way you know it's going to get used instead of it being "shelfwere".* [P15]

Several domain experts did not want to rely on a single threat modeling method and instead created their own by combining and tailoring multiple approaches to align with the organisation and its stakeholders: *"So it would trace each threat described by the brainstorming session using STRIDE. Into a prioritized list of ranked threats using DREAD. And then ..."* [P15] and *"I say: one-size fits all, does not exist. All models have something, but they are never quite complete. Or they never quite cover the target"* [P11] Another common occurrence was that the selection of threat modeling methods was based on perceived popularity, familiarity, or the instinct of the decision-maker. When an interviewee was asked if they would do it differently if they could do it again, the response was: *"No. Even today we would probably start with the simplest model and then let them mature to the point where they find that it is inadequate for their needs."* [P15] Subjective selection often went hand-in-hand with the argument of opting for a lightweight approach and then building quality and complexity over time. All of this, with the primary aim to increase the odds of threat modeling being actually done. However, a couple of experts argued that this adoption becomes less important whenever threat modeling is blocking release and more important when there is mandatory compliance with regulators. One participant stated: *"There are of course some companies ... that fall under heavy external regulators. They also have real requirements ... You can very simply say yes, that (threat modeling) takes me too much time, but then such an external regulator says: Fine, but you are no longer a company either. ... I would especially choose a tool that has the highest possible adoption within the teams."* [P5] In any case, the support of higher management is needed to elevate the practice of threat modeling within an organisation.

7.3 High-over goals

One of the main high-over goals was to increase security awareness. The domain experts identified three levels of security awareness. The first level involves educating developers about the advantages of threat modeling and security practices in general. The main advantage given is that it is expected to save time and resources in the long run. Additionally, neglecting security measures can lead to threats being acknowledged only after they have caused damage. Secondly, awareness in terms of understanding for whom or what you are doing security. By using threat modeling and brainstorming about threats and their impacts on the organisation, the developer is made aware that: *"You do not do it for me as a security officer. You do it to make the company more resilient."* [P5] The third level is about awareness in terms of increasing security knowledge. According to the interviews, threat modeling can be used to help teach developers security-centric thinking. Over time they will use this knowledge in future system design projects.

Therefore, raising awareness is expected to contribute towards achieving the overall security objective of enhancing the organisation's security posture. When this general security goal was mentioned, it was often paired with explaining the expected outcome based on their perceived scope of threat modeling (See Section 7.4). According to the experts, the outcome should contribute to the security posture, for example by discovering new threats that other implemented security measures fail to discover: *"... you want to model a specific architecture and find threats that exist. But threat modeling is just one way to do this ... So a side goal of it would be that the threats identified with the threat modeling tool are unique in comparison with all the other tools that already exist."* [P1]

The last observed goal is to minimize the extra workload for the developers. Security can be very time-consuming. The fundamental strength of threat modeling lies in the speed and low entry barrier for doing it:*"Threat modeling, on the contrary, is a very powerful technique that you can start using fairly quickly and that provides insights fairly quickly"* [P5]. This is reflected in the adoption requirement explained in Section 7.2. Here, the objective does not reflect what is achieved but rather is concerned with actually doing threat modeling.

While several goals and scopes were shared among interviewees, a couple were unique to a few individuals. These specific objectives are most often related to the context of their organisation, such as being able to test compliance: *So if you are at a large company, they will usually have security standards and product security requirements. ... if you are threat modeling, it makes sense to ask if what you are building is compliant with those things."* [P16] Other times it seemed to be caused by having an alternative understanding of the threat modeling concept, based on how their organisation approaches threat modeling. This is best explained with one of the survey comments: *"A good modeling framework should generate more or less copy-paste logic for the detection tool."* [P7] which was further elaborated in the interview: *"The moment you have received threat intelligence about Korea and you have distilled the good TTPs there. Then, in an ideal situation, you can just immediately pass it on to your firewalls and your IDRs."* [P7] This is in essence threat modeling, but this was the first time that the automatic generation of the actual mitigations for newly discovered threats in the threat landscape was mentioned as a goal.

7.4 The goal and scope of threat modeling

Due to the absence of a standardized definition in the literature, a qualitative analysis was conducted to obtain the common goals of threat modeling. This will provide context to what practitioners are expecting and looking for when picking a threat modeling method. The most commonly mentioned goal was to identify relevant vulnerabilities, threats, or risks. In the words of the domain expert: *"The most important thing is to identify your pain points." [P5]*

There is a significant conceptual distinction among these three goals. Although vulnerabilities and threats are different, they both identify the pain points within the system's context. This is what a group of our participants emphasized. Using threat modeling to identify a wide variety of potential pain points, which can then serve as input for risk assessment: *"... then we need to actually think in terms of risk from the beginning, and I think that's a bit of backward thinking when it comes to threat modeling. Because we're trying to find input for risk analysis."* [P17] Another group of experts focused on identifying risks. A threat becomes a risk once that threat is analyzed in the context of the organisation. Hence, they believe that threat modeling should incorporate a form of analysis as part of the process: *"A threat model should serve to allow a good decision to be made of what to address and what not to address."* [P12] When asked about this analysis it is often referred to as a prioritization or relevancy assessment based on a risk calculation (impact x likelihood). This seems to arise from a semantic issue related to the scope of threat modeling.

During the investigation of participants that preferred the separation of the analysis and identification, a common reasoning was found that is rooted in a conflict of interest. Those who perform the threat modeling are often part of the team responsible for the system being modeled. If a threat is assessed as a high risk, the team is tasked with additional work to treat it. An expert explained that *"most of those problems we are talking about are going to need to be fixed by the developers. So you're basically telling them: How to give yourself more work?"* [P16] This human aspect was observed to be a barrier to conducting an unbiased assessment. Furthermore, when analysis and identification of threats are intertwined, users tend to do them together. According to the participants, this leads to threats being disregarded based on perceived relevancy before they are documented, and thus not assessed. When a change occurs in the system or external environment that affects the risk rating, a previously unrecorded threat may become relevant, yet it remains undetected due to the previous assessment. Additionally, by delegating the responsibility of risk rating to users, the organisation exposes itself to the following problem: *"Users may not be aware of other areas in the system. Where the thing they think is low priority could actually be chained with something else. Now it puts the entire system at a higher risk than needed."* [P16] These experiences led to the expert recommendation of clearly separating identification and analysis into two different processes and involving appropriate experts in both processes.

There were a few participants who considered risk treatment to be the main objective of threat modeling. One expert stated: *"After all, it is about risk treatment. So that does not all have to be mitigated. You can also accept things"* [P11]. However, most experts viewed threat modeling as a process of identifying pain points and suggesting potential solutions to manage those risks. It should be noted that most often implementing these solutions was not considered a part of threat modeling. Nevertheless, it was generally emphasized that taking action based on the outcomes of threat modeling must be a high priority: *"So threat modeling should influence subsequent actions. If it does not influence anything, then why did you do it?"* [P2]

7.5 The tool and technique paradigm

To answer SRQ3: *What type of systematic approach can be used for threat modeling method selection?*, a segment of the interview explored the perceived relationship between tools and techniques to establish the context of these concepts. These components were hypothesized to influence the decision-making approach. Therefore, participants were asked to elaborate on the significance of both components. There was no consensus regarding what was more significant, although there seemed to be friction between the concepts. A couple of participants held extreme views, emphasizing the tool's paramount importance over techniques by conveying their needs in terms of tools, even when directly queried about technique or method aspects: *"I would select a specific threat modeling tool... Do I need to think of it within a tool? ... I have an extra: Ease of use, because ... and have the features that are within the tool ... The tool should offer..."* [P1] While others expressed the opinion that techniques were crucial, and the tool's significance was close to negligible: *For me it (a tool) is not important at all. Because I think the core of threat modeling is not about a tool. It's a way of thinking about what you're doing. ... I think it (the importance of a technique) is a ten. If it's too complex and everything we discussed already, it means it's not going to get adopted.* [P16]

Upon further investigation into the cause of this observation, an expert who had implemented threat modeling in multiple organisations stated the following when asked about the importance of a tool: *"So tool support is a good thing, but not something I'd really consider in the selection unless there is already an established method. In which case I would be going out and looking for a tool that would match that one."* [P13] Upon closer examination, the situations in which these extreme views occurred matched the context described in the quote. The experts who were working in a large organisation with a relatively high security maturity level were the sole cause of the extreme tool preference. On the other hand, practitioners that did not find a tool very important expressed that most often in the context of lacking an established threat modeling basis. In the latter case, it was frequently recommended to start threat modeling without a tool and subsequently look for a tool that aligns with the adopted threat modeling approach (See previous quote of P13). Furthermore, there were participants who were dissatisfied with the currently available tools and, as a result, leaned towards prioritizing techniques over tools: *"I looked at that (Microsoft TMT) and did not find it very useful ..., so I do believe in it as ... More like drawing a picture and then brainstorming ... It could be a bit more structured, ... and then you do need tools, but they should mainly be supportive."* [P12]

The tool and technique paradigm demonstrates that the organisational context and individual experiences have a major impact on how experts perceive and choose threat modeling methods. To create a comprehensive approach that caters to different viewpoints and meets specific requirements in different situations, it is crucial to be aware of these nuances.

7.6 Requirements from practice

To contribute to SRQ2: *What are the criteria for selecting threat modeling approaches?* additional requirements were identified through conducting the interviews. In total, 29 main requirements were added and 14 new sub-user stories were discovered. The majority of the newly uncovered requirements were associated with tools or the process supporting the method. Moreover, during the interview, all the requirements that had been previously discovered through the survey were mentioned by the participants. All requirements found during the interviews can be found in Appendix D.1.

A couple of requirements stood out due to their context. According to the experts, the output of the threat modeling method is crucial. To illustrate, during a discussion about how threat modeling should be conducted, one of the interviewees emphasized: *“Whether it comes out of an automated tool or whether it comes out of spreadsheets makes no difference. As long as it produces and provides that kind of information they (the developers) need to do something about it.”* [P15] This sentiment was observed multiple times among various participants, particularly when discussing their expectations of the threat modeling method. This finding led to the formulation of requirement US206 - Actionable findings. Instead of focusing on specific attributes of output, such as the format of the outcome, practitioners were more concerned about the practical usability of the results by stakeholders. This perception is also reflected in US205 - Security patterns: *"As a decision-maker, I want the method to assist with identifying security patterns, so I can identify security areas that need to be fixed."* There are various potential implementations of this requirement, each with its own strengths and weaknesses. However, the study did not have sufficient resources to explore these requirements in detail, so they are kept in their general sense.

In addition, numerous requirements were explained in the context of their organisation, more details regarding this dependency can be found in Section 7.4. Notably, decision-makers frequently framed their statements within the context of their respective organisations, with a particular emphasis on users and other stakeholders. This observation is in line with the survey results regarding the relative ranking of quality criteria. Consider, for instance, US810 - medium familiarity: *"As a decision-maker, I want the method to match the context of the stakeholders, so they can easily use it."* In this case, stakeholders refer to core users such as source code developers, as well as, individuals without technical knowledge or even the expert themselves. Even within these stakeholder groups, there is a huge difference in context between individuals in terms of familiarity and expertise. These non-functional requirements demand additional context beyond the method itself to evaluate their satisfaction properly. As a result, they should be dealt with accordingly when used in the decision model.

7.6.1 Top requirements in practice

To understand what requirements are important for selecting threat modeling methods, this study conducted an analysis of the frequency of mentions across all transcripts. These results were subsequently used to determine the final set of requirements for the decision model as explained in Section 8.2. It is worth noting that the occurrence analysis was carried out on requirements derived from the SLR, as well as those discovered during the survey and expert interviews. The latter are indicated with (New). For detailed results of the requirement occurrence analysis, refer to Appendix C.1.

The most frequently mentioned requirement was automation (US805). Ten out of the twelve domain experts agreed that some type of automation would be useful in threat modeling and that this could be a factor in selecting a method. Despite several experts indicating that having a tool was not their primary focus when selecting a threat modeling method, they acknowledged the benefits of automating certain steps: *"So tool support is a good thing, but not something I'd really consider in* *the selection ... Ideally the tool is there to automate the bits of the technique that waste people's time".* [P13] Four types of automation were mentioned: automatic threat generation, automatic system representation generation, automatic prioritization based on user input, and automatic effect propagation.

Additionally, input type (US102), threat generation (US601), and pipeline integration (US808) were each brought up by nine experts (75%). Pipeline integration was a newly discovered requirement during the interviews. To illustrate, one participant said: *"I have started using Plant-UML more. Because ... and it is quite nice to actually have the diagrams and the model in a format that you can render in the pipeline. So, they (the developers) can keep the data together with the code, modify it, have its version control, and stuff like that."* [P17] The frequency of this requirement indicates that experts consider it crucial for a tool to be integrated into a pipeline rather than functioning as a standalone concept. Table 7.1 displays the five most frequently mentioned requirements sorted by unique mentions over all interviews.

7.7 Quality criteria from practice

The results of the interviews also enhanced the set of quality criteria. The following qualities were discovered during the interviews and positioned in the quality hierarchy based on their respective descriptions:

- **Uniformity:** The threat modeling method is consistent, standardized, or homogeneous across different teams or departments.

Table 7.1: The top five most mentioned requirements by domain experts. The Mention rate denotes the unqiue expert mentions. The total mentions indicate the number of times the requirement has been mentioned in general. The complete results of the requirement occurrence analysis can be found in Appendix C.1.

ID	**Requirement**	**Description**	**Mention rate**	**Total mentions**
US805	Automation	As a decision-maker, I want to have a tool that assists the threat modeling technique by (semi-) automating some of the steps.	83.33%	36
US102	Input type	What type of input is required in order to start the threat analysis.	75%	30
US601	Threat generation	The process of identifying and establishing a list of potential threats.	75%	29
US808	Pipeline integration	As a decision-maker, I want the tool to be integrated into a pipeline, so it will be integrated with the other tools.	75%	21

- **Accuracy:** Does the method provide results that correctly represent reality.
- **Effectiveness:** Are the outcomes of the method implemented and effective in treating the threat.
- **Discoverability:** The ability to discover the results from threat modeling. (e.g., How do people know that the team has done threat modeling? How do people know where to find the results?)
- **Organisational fit:** Compatibility or alignment between the threat modeling method and the organisation in terms of goals, culture, and employee skills.
- **Usefulness:** The capacity of the threat modeling method to be beneficial, valuable, or helpful in achieving the desired purpose or goal.

To better fit the practical context, certain qualities were adjusted. "Grokkiness", was changed into “clear process” while maintaining the definition. Similarly, "Non-ambiguity" has been changed to “repeatability”. "Compatibility with related processes" has been extended to include tools, due to practitioners often explaining this quality using the software solution that supports the process. Furthermore, "outcome" has been incorporated into the definition of "suitability" to capture the observation that expectations are often communicated in terms of results. Besides these, several minor changes have been made. The refined hierarchy of qualities can be found in Appendix D.3.

7.7.1 Contextual highlights

During the interviews, participants provided valuable context surrounding the qualities. Although it is not possible to report on all of it, these are the most insightful highlights.

Software evolution support. There was a degree of confusion among participants regarding the subcomponents of software evolution support. For example, when prompted to explain "change propagation" the response was: *"... When something changes, we need to be able to see how the system or model evolved. So, we can look at the first version and compare it to the latest version. Then we check: Do you have anything between those two that has changed in terms of threats?"* [P17] What is described here is change impact analysis, a different subcomponent of software evolution support. So, participants did not fully comprehend the distinctions between these subcomponents. This was also confirmed in a comment of the survey: *"I was struggling to feel the difference between "modularity" and "component architecture"* [P2] Specifically, modularity and component architecture were interpreted as closely related, as were change propagation and change impact analysis.

While discussing general important factors, modularity, and change impact analysis, were emphasized as important. For example, modularity was discussed as *"What I think is important about this, ... How can you break this down into different target groups?* [P11] The examples provided by the interviewees were not explicitly related to software evolution support. Instead, the discussions revolved around the terms in a general context. When change impact analysis was discussed, it resolved around general change resilience: *"He (the threat model) actually needs to be constantly updated. The world of cybersecurity is constantly experiencing change. Hence, you actually have to constantly update all your techniques just like that.* [P7] Change resilience is not restricted to handling an evolving system but also includes changes in the organisation and the dynamic threat landscape. Interestingly, it appeared that the demand for a method that was able to handle changes in the threat landscape was even more prominent than something that could deal with an evolving system.

Cost. Cost was often mentioned as one of the organisational aspects that influence the method selection. To quote a couple of experts: *"Budget, of course. It's always someone who has to pay for it."* [P13] or *"A threat model - as I see it - is quite labour-intensive so it does cost a lot. ... So cost is definitely also important.* [P12] The term cost was discussed in various forms, including tool and labor costs, as well as, the potential cost of a security incident when threat modeling was not employed. Regardless of the type of cost discussed, the interviewees considered it in the context of a cost-benefit analysis. The benefit most commonly mentioned was the perceived usefulness of the threat modeling outcome for the organisation. When asked to elaborate on how this analysis was conducted, the interviewees explained that it was a qualitative assessment from their perspective. One expert offered a compelling insight into the challenge of doing this analysis, stating: *"The business case is difficult to make, because what are the costs of an incident? Well, if you ask us security or a business risk management person, of course, you get big numbers Whereas if you ask a developer, it is such small numbers. So, what is the truth? It has never happened, so we do not know until it happened with the competitor. They lost 10 million, do you want that? Those are very difficult discussions to substantiate quantitatively."* [P5]

Documentation. In the literature the quality of documentation is expressed in terms of the method documentation being extensive and clear. Although this definition was presented, during the interviews, documentation was often perceived differently, referring to the documentation of the threat modeling artifact (US607): *"Microsoft TMT is very much geared towards the components they know. You drag anything and everything into that, you have to fill in a lot of details on them,* *but that does help with good documentation."* [P11] A notable discrepancy between the literature and practice emerged when discussing the documentation of the method. While the literature mentions extensive documentation as a benefit, the experts preferred clear yet concise documentation. While the former is useful for novices, for the majority the latter is more important. Having only extensive documentation can have a significant adverse impact on the adoption: *My number one goal is that some form of threat modeling is actually done. ... The things that have 300 pages books, I think that's already a failure there. My point is: "How can I get developers to actually start threat modeling?"* [P12]

Organisational fit. Although it was newly added during the interviews, it was mentioned by all interviewees. Its relevance was conveyed through relative statements, such as: *"Free form is too woolly, and I do not think it is a sensible choice for our levels of maturity."* [P2] Practitioners frequently referenced security maturity or other organisational aspects to highlight the importance of organisational fit in the selection: *"I meant how mature they are in continuous security work and risk assessments, because if they have a quite high maturity level then we can take a deeper and more complex method instead of something that is simpler and just scratches the surface."* [P17] and *"If it's an organisation where they do those large weekly meetings, the monolithic method works better because you can just integrate it with. If they're more, agile-focused with five-minute stand-ups each day. You need that thing that's more integrated."* [P13]. The definition of organisational fit should be interpreted broadly, encompassing various aspects, including dealing with regulatory compliance, which can indirectly shape the organisational culture.

7.7.2 Top quality criteria in practice

In order to determine the significance of qualities in selecting threat modeling methods, the frequency of mentioned qualities was analyzed across all transcripts. Two qualities consistently emerged in all interviews: *Organisational fit* and *usefulness*. These qualities were not only mentioned by all experts but also prominently repeated throughout individual interviews. Additionally, *clear process*, *ease of use*, and *understandability* were brought up in eleven out of twelve interviews, highlighting their importance. Among the top five most important quality criteria, three of them were newly found during expert interviews or surveys. Table 7.2 shows the most mentioned criteria sorted by the mention rate, i.e. the unique mentions divided by the total interviews. The full results of the quality occurrence analysis can be found in Appendix C.2

TABLE 7.2: The top five most mentioned quality criteria by domain experts. The Mention rate denotes the unqiue expert mentions. The total mentions indicate the number of segments in which the quality has been detected. The complete results of the quality occurrence analysis can be found in Appendix C.2.

Quality criteria	**Definition**	**Mention rate**	**Total mentions**
Usefulness	The capacity of the threat modeling method to be beneficial, valuable, or helpful in achieving the desired purpose or goal.	100%	45
Organisational fit	Compatibility or alignment between the threat modeling method and the organisation in terms of goals, culture, and employee skills.	100%	40
Clear process	Intuitiveness of the threat modelling method for the stakeholders intended to apply it.	91.67%	34
Ease of use	Simplicity of use of the technique, tool and notation.	91.67%	30
Understandability	How easy is it is to understand the outcome of the method.	91.67%	19

7.8 Relationships between requirements and quality criteria

The other type of interview performed was the relationship interview. Four interviews of this type were successfully conducted with domain experts that had a background in teaching cyber security to students or participants that showed a greater understanding due to having done research dedicated to threat modeling (See Section 5.1.2 for details). The participants were prompted to engage in brainstorming focused on aspects that could be included in a threat modeling method to positively contribute towards these qualities.

Using the same steps and codebook formed in Section 7.1.1 the interview transcripts were coded. The only difference between coding these interviews and the requirement validation and exploration interviews, is that the structure and goal of the interviews required dropping the decision that only codes were assigned whenever they were naturally brought up by the interviewee. As a result, these codes are used solely to establish positive relationships and are not subjected to any frequency analysis.

Establishing relationships between the qualities and features has been done by combining the co-occurrence analyses between the qualities and requirements of both types of interviews. These were combined using an OR rule. Whenever a general code was applied for a main requirement. The results from that co-occurrence analysis were applied to all its sub-user stories. Table 7.3 shows the

Table 7.3: A subset of the quality x requirement mapping. A 1 indicates that there is at least one co-occurrence was found between the quality and requirement and a 0 indicates there has not been any co-occurrence found. Some qualities have been shortened, due to readability. The full mapping can be found in Appendix C.3.

	Applicability									**Efficiency**					**Reliability**															
Quality x Requirement Mapping	Accuracy	Clear process	Comp. with Agile..	Comp. with non-agile..	Comp. with related..	Organisational fit	Usefulness	Ease of use	Learning curve	Cost	Reusability	Scalability	Tailorability	Uniformity	Completeness	Discoverability	Technique documentation	Tool documentation	Effectiveness / testability	Technique maturity	Tool maturity	Repeatability	Change impact analysis	Change propagation	Component architecture	Modularity	Suitability	Modifiable	Traceability	Understandability
Actionable findings (US206)	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0	1
Analysis level (US705): System architecture analysis (c)	0	1	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Applicability to system (US503)	1	0	0	0	1	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
Approach (US301): Attack-centric (a)	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Approach (US301): Risk-centric (b)	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Approach (US301): Software-centric (c)	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Artifact documentation (US607)	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1
Automation (US805): Automatic effect propagation (d)	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1	0	1	1
Automation (US805): Automatic prioritization (c)	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	1	0	0	1	1	1	1	1	1	0	1	1
Automation (US805): Automatic system representation generation (b)	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1	0	0	1
Automation (US805): Automatic threat generation (a)	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	0	1	0	0	1	1	1	1	1	1	0	0	1
Change highlights (US824): Common control change (d)	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
Change highlights (US824): Organisational change (b)	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
Change highlights (US824): Prioritizatoin change (c)	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
Change highlights (US824): Threat input change (a)	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
Classes and instances (US818)	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Common vulnerabilities (US207)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Component templates (US819)	0	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0
Continuous threat modeling (US609)	1	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	0	0	0

identified relationships based on this analysis. A 1 indicates that there is at least one co-occurrence found between the quality and requirement and a 0 indicates there has not been any co-occurrence found. Note, the Q x F mapping presented here is updated for the final set of requirements after the modifications explained in Section 8.2. Moreover, for a couple requirements the interviews were not sufficient in providing a relationship with quality, these are highlighted in red. Due to the explorative nature of creating this mapping, it was chosen not to increase the co-occurrence threshold.

Although the quality attributes and mapping between requirements have been collected, they have not been used as part of the decision model. Future research can refine these relationships and use them as part of their own decision model to allow a more comprehensive evaluation. To provide additional insights, participants highlighted the existence of additional relationship types that are not encompassed within the current framework. For instance, there are relationships among the quality criteria that lead to a significant overlap in contributing requirements. As stated by an experts: *"Grokkiness (clear process) is almost the combination of learning curve and ease of use.* [P13] and *This (completeness) is also related to reproducibility (repeatability), I think. The more complete the knowledge is, the easier it will be for other people to arrive at the same result, using the same method.* [P8]. When specifically asked about requirements that contribute to the *learning curve* and subsequently inquiring about the same for *ease of use* a participant answered: *"I mean it is like almost the same"* [P10] This was also explicitly agreed upon by three out of four participants: *"It is all those little things that are also related to ease of use, that are ultimately going to lower the learning curve."* [P8] The actual representation of these relationships is not perceived as a hierarchical structure with binary relationships as employed in this study. To illustrate this, a participant mentioned: *"I am very curious about that whole lettuce of combinations of qualities and influence."* [P8]

8 Designing the systematic approach

As discussed in Section 3.3, the systematic decision-making approach is inspired by the MCDM framework of Farshidi [51]. The method is centered around having a decision model that is initialized as a decision-support system (DSS). This section discusses the proposed approach, outlining the steps taken, and the decisions made in the process of turning the findings from the systematic literature, expert survey, and interviews into a systematic decision-making approach. This finalizes the answer to SRQ3:

What type of systematic approach can be used for threat modeling method selection?

8.1 The decision model

Like the framework of Farshidi [51], this approach aims to guide practitioners by offering support for selecting the most appropriate alternatives based on specific domain requirements. In this study, the alternatives are threat modeling methods. These are more complex than what is regularly utilized in the framework because threat modeling methods consist of three subcomponents; Tool, Technique, and notation. We decided to use a simplified version of the decision model in this study, due to the multiple dimensions considered and resource constraints. Figure 8.1 illustrates the components of the model.

Our decision model utilizes a set of refined requirements. We use and extend the Boolean data type to consider partial supportability, based on the process dependency of the requirement. The weighted sum method has been deployed in order to calculate the score. Additionally, there are some specific non-functional requirements, that need context beyond the method itself to test their satisfaction. This study proposes to use an experimental global multiplier to deal with these requirements.

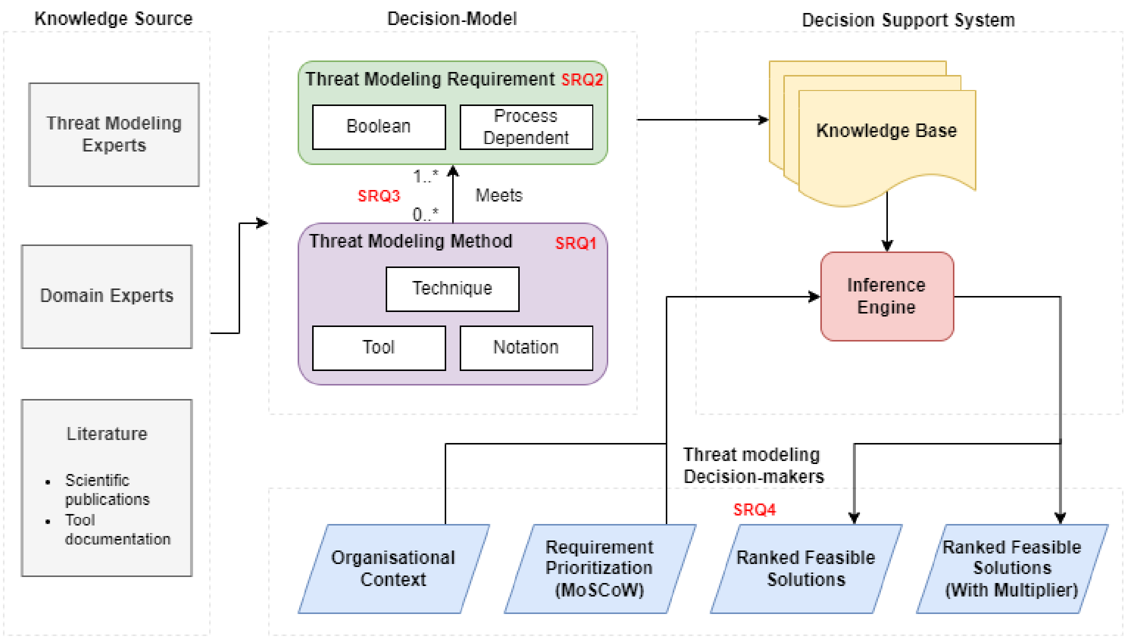

FIGURE 8.1: An illustration of the main components of the decision-making framework adapted from [51]. The research questions are marked in red.

The DSS receives the subjective preferences of the decision-maker as input in terms of threat modeling method requirements that are prioritized using MoSCoW. We adopted MoSCoW since decision-makers already prioritize their domain requirements using this framework [32]. This prioritization will be used as individual weights on each requirement in terms of $\omega MoSCoW = \omega Must$, $\omega Should, \omega Could, \omega Won't$. There is a hard constraint on requirements with a *Must have* and a *Won't have* priority. Meaning, the DSS marks every alternative that either does not support a requirement prioritized with *Must have* or does support a requirement that is prioritized with a *Won't have* as an infeasible solution. Infeasible solutions are not considered as suitable alternatives and therefore will not be considered. Subsequently, the DSS calculates a score for each feasible alternative based on the weighted sum of the requirement prioritization.

The weights for each prioritization are by default $\omega M = 100$, $\omega S = 50$, $\omega C = 25$, and $\omega W = 0$.

This is roughly in line with the commonly used 60-20-20% approach used for development budget allocation with MoSCoW [31]. The weights for *Must have* are kept around 60% of the budget, while *Could have* has a slightly lower weight and *Should* has a higher weight. This deviation is grounded in the high relevancy rate of the requirements measured in the survey. Therefore, it is not expected to have any statistical implications [120]. Based on the assigned scores, the suitable alternatives are arranged in descending order to create a prioritized list of potential solutions.

8.2 Post-interview requirement refinements

Throughout the research project, several iterations were undertaken to elevate the list of requirements and qualities to the next level. The SLR provided requirements from the literature. Before the interview, they were refined to fit the context of the decision-maker. During the interview, some additional requirements and qualities were identified. However, it was made clear that the presented requirements were still very multi-interpretable and often contained different terminology than commonly used in practice. Additionally, multiple examples were discovered to be influential in shaping the interpretation of the requirements. While working towards creating the decision model based on these requirements, it was decided to once more refine the concept based on the terminology used by the domain experts in our interviews.

Besides re-writing the requirements, we added and moved around several requirements from the "other" category to a more appropriate epic. Furthermore, all requirements and sub-user stories that were found relevant or applicable for less than their average were checked. Sections 6.2 and 6.3 provided a starting point for refining the requirements. This was supplemented with the findings of the interviews. We included newly discovered requirements that were mentioned by at least two participants. Further modifications were made based on qualitative interpretations as well as the occurrence analysis reported in Chapter 7. The changes have all been logged and explained in Appendix D.2.

Some requirements were deemed relevant but were not used in the final decision model. This was due to their high dependence on the context or in case they did not seem to have a significant benchmark in the context of a tool, technique, or notation. For example, US605 – Trigger: *When the threat modeling method is initiated.* This could be dynamic when threat modeling is initiated based on changes or static based on a time interval. Although some methods are centered around being continuous, both types of triggers could be used in combination with any threat modeling method. So, it does not really help in selecting a method, it is almost fully reflected in how the organisation implements the method. Finally, 95 requirements were considered as part of the score calculation.

8.3 Benchmarking alternatives

The next step towards creating the decision model is benchmarking the alternatives. This was split up into three different steps. First, a set of threat modeling methods was selected. Then each subcomponent was benchmarked based on the appropriate requirements. Lastly, the benchmarking of the subcomponents was combined and re-evaluated to obtain a final mapping consisting of a set of benchmarked threat modeling methods.

8.3.1 Selection of alternatives

The alternatives were selected based on the results of the systematic literature review, this can be found in Section 4.6.1. A total of 63 potential techniques were identified, in the SLR and individually examined. A subset of techniques was selected based on the inclusion and exclusion criteria (See Table 8.1). Note that these were not used as hard constraints but rather as a collective evaluation. Meaning that a technique was rated more relevant when they meet an inclusion criterion and vice versa. Finally, we decided to consider thirteen different techniques.

In the SLR we identified twenty potential tools but the majority of these tools were based on a paper and did not have publicly accessible documentation or software. Therefore, six of the most commonly available tools have been selected that were also mentioned in the interviews with domain experts. Additionally, a significant number of techniques can be employed without being tied to a particular tool. For example, when the technique creates a textual artifact, such as persona non grata. In other instances, when relying on graphical methods, a simple drawing tool can be used such as Drawio and MS Visio, or even a non-digital tool such as a whiteboard. Therefore, two additional tool options were added: *“No tool”* and *“Simple drawing tool”*, to represent these cases.

Lastly, the notations have been selected based on the subset of techniques considered. Six different notations have been considered of which DFD is the most common one. Furthermore, there are cases where techniques did not specify a particular graphical notation or explicitly employed a textual representation. These have been dealt with by introducing the option called "Non-specific or textual".

TABLE 8.1: Evaluation criteria for creating a subset of technique components for the decision model.

Inclusion criteria
Techniques that have combined paper citations over 100.
Techniques that have multiple independent papers.
Techniques that were mentioned by practioners in an interview.
Exclusion criteria
Techniques that have less than 100 combined paper citations.
Techniques that rely on a tool, which cannot be found publicly are disregarded.
Source code-based techniques are disregarded due to requirement scoping.
Requirement engineering based techniques are disregarded due to requirement scoping.
Techniques that are only relevant for a specific Cyber-Physical Systems or for control systems. (System that control other systems)
Techniques that combine multiple existing models.

8.3.2 Benchmarking subcomponents

After all alternative method subcomponents have been selected, each component is benchmarked individually, based on the set of refined requirements and sub-user stories that can be found in Appendix D.2. Most of the requirements are only relevant to one or two components. A component is either a tool, technique, or notation. An assessment of relevancy has been made in which we rated each requirement based on the method component(s) that can be used to benchmark it. To illustrate the results of this assessment, a subset is shown in Table 8.2. Besides the tool, technique, and notation, a column has been added for the process. When talking to domain experts, it became evident that some of the requirements are partly based on the structure of the surrounding process. Meaning that there are requirements that are not explicitly supported by the method but are still applicable when the method is implemented in a certain way. This finding is referred to as partial supportability.

To evolve the decision model, we decided to differentiate between requirements that are not process-dependent and requirements that are. Therefore, the collection of requirements considered in this study can be denoted as $Requirements = Requirements^B \cup Requirements^P$. The requirements that are not process dependent are benchmarked using the standard Boolean data type, where $BRC : Requirement^B \times Components \rightarrow \{0, 1\}$. Requirements that are process dependent use an extension of this data type by adding accountability for partial supportability, where $PRC : Requirement^P xComponents \rightarrow \{0, 0.5, 1\}$. Both mappings denote the supportability of the

Table 8.2: A subset of the relevancy assessment. Checkmarks indicate that the requirement is assessed based on the component. Cross marks signify the component is not relevant for benchmarking the requirement. The full relevancy assessment can be found in Appendix E.1.

ID	Name	Tool	Technique	Notation	Process
US102	**System input**	No assessment needed			
b	Security assumptions	✗	✓	✗	✓
c	System goals	✗	✓	✗	✓
US201	**Output type**	No assessment needed			
a	Security threats	✗	✓	✗	✗
b	Threat mitigations	✗	✓	✗	✗
US301	**Approach**	No assessment needed			
a	Attack-centric approach	✗	✓	✗	✗
b	Risk-centric approach	✗	✓	✗	✗
US401	**Notation Abstraction**	No assessment needed			
a	Goal-centric	✗	✗	✓	✗
b	Model-centric	✗	✗	✓	✗
US501	Knowledge barriers	✓	✓	✓	✗
US502	**Layer**	No assessment needed			
a	Software layer	✗	✓	✗	✗
US601	**Threat generation**	No assessment needed			
a	Methodology-based threat generation	✗	✓	✗	✓
b	User-based threat generation	✗	✓	✗	✓
US702	Effect propagation	✗	✓	✗	✓
US704	Threat library	✗	✓	✗	✓
US803	Validation	✓	✗	✗	✗
US804	Verification	✓	✗	✗	✗

requirement by one of the subcomponents of a threat modeling method. Where $(B/P)RC(r,c) = 0$ when the component does not support the requirements and $(B/P)RC(r,c) = 1$ is used in the case that the requirement is supported by the component. Furthermore, $PRC(r,c) = 0.5$ demonstrates the partial supportability of process-dependent requirements.

The mapping of the requirements to the techniques is achieved through a brainstorming session with a domain expert and subsequently resolving uncertainties by consulting the papers associated with the technique. The mapping of requirements to tool components is based on the tool documentation and our expert interviews. The last mapping, between the requirements and the notation, is accomplished by examining the contents of the notation.

8.3.3 Combining benchmarks

After all method components were benchmarked, the components were combined into the final alternative methods, where $Method = Component^{Tool} \cup Component^{Technique} \cup Component^{Notation}$. These are combined based on the application context of the components. For example, when discussing STRIDE, it can be supported by either Microsoft TMT, OWASP Threat Dragon, or Iriusrisk, or it can be done using a simple drawing tool. In any of these cases, the DFD notation is the basis for creating threat models. Given these observations, four threat modeling methods (alternatives) are considered. As a result, eighteen different threat modeling methods are considered in the decision model. (See Table 8.3). The method that contains pyTM as a tool is not included in the final decision model. While looking at the context of the tool, pyTM does not utilize a specific threat modeling technique to maintain flexibility. Hence, it does not align well with the benchmarking approach adopted in this study.

The requirement benchmarks related to the components of each method were concatenated. In some instances, requirements were benchmarked by multiple components. For example, when

Table 8.3: The final threat modeling method alternatives and their components.

ID	Technique	Tool	Notation
A1	STRIDE	Microsoft TMT	DFD
A2	STRIDE	OWASP threat dragon	DFD
A3	LINDDUN	OWASP threat dragon	DFD
A4	STRIDE	Iriusrisk	DFD
A5	LINDDUN	Iriusrisk	DFD
A6	VAST	Threatmodeler	Process flow diagram
A7	STRIDE	Threagile	DFD
A8	Attack trees	Simple drawing tool	Node-edge
A9	Defense trees	Simple drawing tool	Node-edge
A10	Attacker scenarios	No tool	Non-specific / textual
A11	CORAS	Simple drawing tool	CML
A12	STRIDE	Simple drawing tool	DFD
A13	LINDDUN	Simple drawing tool	DFD
A14	P.A.S.T.A	No tool	Non-specific / textual
A15	Misuse patterns	No tool	Non-specific / textual
A16	Petri-nets	Simple drawing tool	Petri-nets
A17	Fault-tree analysis	Simple drawing tool	Fault-tree notation
A18	Persona non grata	No tool	Non-specific / textual

the relevancy assessment identified that a requirement is relevant for both a technique and a tool. In case that happened, the highest benchmarked support was assigned for that method alternative. To illustrate, US202a – High-level outcome, is assessed based on the tool and technique. Also, this requirement has a process dependency, which allows for $PRC(r,c) = 0.5$. In case that $PRC(US202a, tool) = 0.5$ and $PRC(US202a, technique) = 1$ this is combined into $PRC(US202, method) = 1$. A subset of the resulting mapping between the requirements and alternative threat modeling methods can be found in Table 8.4.

Table 8.4: A subset of the requirement and alternative mapping. A zero indicates that there has been no evidence that the requirement is met by the alternative. A 0.5 indicates partial supportability. A one indicates that the requirement is met by the alternative. The complete mapping can be found in Appendix E.2.

ID	Placeholder	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18
	E001: Input																		
US102	**System input**	**No benchmark needed**																	
b	Security assumptions	1	1	1	1	1	0,5	1	0,5	0,5	0,5	0	1	1	1	0,5	0,5	0,5	0,5
c	System goals	0,5	0,5	0,5	0,5	0,5	0,5	0,5	1	1	0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0
d	Architectural design	1	1	1	1	1	0,5	1	0,5	0,5	0	1	1	1	1	0,5	0,5	0,5	0
f	Business processes	0,5	0,5	0,5	0,5	0,5	1	0,5	1	1	0,5	1	0,5	0,5	1	1	1	1	0,5
g	Descriptive language	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
US103	**Threat input**	**No benchmark needed**																	
a	Brainstorming	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
b	Historical data	0,5	0,5	1	0,5	1	1	0,5	0,5	0,5	1	1	0,5	1	1	1	1	1	1
c	Threat intelligence tooling	0,5	0,5	1	0,5	1	1	0,5	0,5	0,5	1	1	0,5	1	1	1	1	1	1
h	Threat actors	0,5	0,5	1	0,5	1	1	0,5	0,5	0,5	1	1	0,5	1	1	1	1	1	1
g	Framework	1	1	1	1	1	0	1	0	0	0	0	1	1	1	0	0	0	1
	E002: Output																		
US201	**Output type**	**No benchmark needed**																	
a	Security threats	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
b	Threat mitigations	1	1	1	1	1	0,5	1	0,5	0,5	0,5	1	1	1	0,5	0,5	0,5	0,5	0,5
c	Security requirements	0,5	0,5	0,5	0,5	0,5	1	0,5	0,5	0,5	0,5	0,5	0,5	0,5	1	0,5	0,5	0,5	0,5
US202	**Output granularity**	**No benchmark needed**																	
a	High-level granularity	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
b	Low-level granularity	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0
US203	**Report representation**	**No benchmark needed**																	
a	List	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
b	Figure	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
US205	Security patterns	1	1	1	1	1	1	1	0	0	1	0	1	1	0	1	0	0	1
US206	Actionable findings	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
US207	Common vulnerabilities	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0

8.4 The final approach

As mentioned before, the approach is centered around having a decision model that is initialized as a decision-support system (DSS). However, as the research progressed and particularly during the interviews with practitioners, it became evident that a substantial portion of the selection is influenced by the stakeholders' perception of threat modeling. As explained in Chapter 7, threat modeling is associated with a variety of goals, scopes, and preferences toward individual components. This is in line with the findings of Xiong and Lagerström [200], which conclude that there is no common definition.

Although this could be seen as a limitation, we decided to use some of the contextual findings from the interviews with domain experts and embed them into the final selection approach. The decision-makers are asked to fill in a couple of preliminary questions, which determine how the decision should be made.

Step 1: The decision-makers are presented with questions regarding the current organisational context. First, the question is asked who the expected users are and what threat modeling technique and notation they are familiar with. Having familiar aspects contributes to the adoption of the threat modeling method, according to our expert interviews. This familiarity context in combination with the prioritization of the respective requirements (US901b and US403) determines whether and which methods should be assigned a familiarity multiplier. The size of the multiplier is 0 to 0.2 based on the prioritization, where $M = 0.2$, $S = 0.1$, $C = 0.05$, and $W = 0$. The multiplier is applied to methods that have the selected technique or notation as a component. Given the experimental nature, the score with and without multipliers are presented independently in the final DSS. Combining the requested context with the prioritized requirement to determine a global multiplier is our approach toward dealing with non-functional requirements that need context beyond the method itself.

Step 2: The decision-maker indicates the groups of requirements they would like to consider. Thirteen statements like *"I want to consider..."* are presented, allowing the decision-maker to respond with either *Yes* or *No*. Each statement applies to a group of requirements and determines which requirements the decision maker is recommended to prioritize. The set of statements and their corresponding requirements can be found in Table 8.5. As a follow-up question, the decision-maker is asked to select and rank the most important requirement groups. At least one and a maximum

TABLE 8.5: The set of statements and their corresponding requirements. These statement could be answered with Yes or No. When a Yes is indicated, the requirements associated are recommended to prioritize.

Statement (I want to consider...)	**Requirements**
... the type of system representation that is required in order to do threat modeling.	System input (US102)
... what data is used as basis for threat identifcation.	Threat input (US103)
... requirements related to the outcome of the threat modeling method.	E002: Output (US201-207)
... the point of view from which threat modeling is done.	Approach (US301), Focus (US302)
... the level of detail on which the system is represented in the model.	Modeling view (US304)
... aspects of the modeling notation which is used to do threat modeling.	E004: Modeling notation (US401-403, US407-409)
... aspects of the modeling notation which are relative to the organisation.	Notation familiarity, Similarity with software specification languages, Notation extensibility (US404-406)
... the system & stakeholder context in which the threat modeling method can or cannot be applied.	E005: Modeling context (US501-506)
... requirements that apply to the process of doing threat modeling in the organisation.	E006: Method process (US601-611)
... the security coverage and depth of the threat analysis.	Effect propagation (US702), Analysis level (US705), Security objective(US706), Threat analysis type (US710)
... how the relevancy of threats is assessed in the context of the system.	Threat library (US704), Prioritization (US707), Risk (US708), Resource valuation (US711), Current security (US712)
... tool features and automation.	E008: Tools (US803-831)
... how threat modeling as a concept is part of the organisation.	E009: Other (US901-903)

of 5 requirement groups can be selected. These determine the starting order in which the requirements are prioritized. Starting off with the most important group of requirements related and then guiding the decision-maker through that list.

During the expert interview, the domain experts remarked that the survey was exhausting, due to the significant number of requirements questioned. After the survey, the set of requirements increased, because of the practical additions. By only considering the most relevant requirement groups, a lightweight variant of the structured selection is proposed, in which only the requirements related to the aspects that matter to the decision-maker are considered in the DSS. Note that the practitioner is encouraged to prioritize beyond this selection to refine the results. It can still occur that some requirements are perceived as non-relevant these should not be prioritized by the decision-maker.

In the case of a group decision, a discussion session is initialized in which inconsistencies in the preliminary questions and prioritization are discussed until a consensus is reached. The final prioritized requirements are put into the DSS which will calculate a score for each feasible alternative as explained in Section 8.1. This results in a ranked list of feasible alternatives. The decision-maker should then take into consideration the feasible threat modeling methods with the highest scores and make a determination on which one to adopt.

9 Case study validation

This section presents the results of the empirical validation conducted. The goal of the case study was to validate the current approach by enrolling it in a practical environment. This will provide an answer to SRQ4:

How well is the proposed decision-making method perceived by the stakeholders?

The approach is tested in the context of a large enterprise. The decision-making approach as described in Section 8.4 has been enrolled for a group of information security officers, responsible for giving advice on security matters.

9.1 Case study context

The participants had between three and thirty years of work experience and worked in the organisation between six months and four years. Their knowledge of threat modeling as measured by their knowledge of techniques, notation, and tools, varied from one to six instances, with three of these components currently being used.Threat modeling was already implemented in their organisation. However, this was negatively received by some of the users, so they were looking to see if there is another threat modeling method that would fit better in their context. More specifically, their focus was directed towards a tool, since a tool was already currently utilized. They classified their IT process maturity using COBIT in between level 2 (managed) and level three (established).

9.2 Qualitative feedback and observations

During the session, we collected feedback regarding our approach and its components. The participants suggested that a significant number of preliminary statements, used to determine which requirements were relevant, were open to multiple interpretations. Rather than giving a definitive

Yes or No, most of the time it was dependent on what was meant by the statement. So, the abstraction level did not match the context of the decision maker. Although the requirements have been refined in multiple iterations, the participants found it challenging to interpret them. It again was suggested that they were multi-interpretable. They illustrated this by providing an example of their specific preferences and requirements, both in terms of what they desired and what they wished to avoid. This was based on the context of the information provided to them by their primary users.

Based on both suggestions we conclude that the content of our approach needs additional clarification. One of the goals of the approach was to be able to support the selection of a threat modeling method without needing too much knowledge of threat modeling. This goal has not been achieved in this study. The abstraction level at which requirements are presented requires explicit knowledge of the common terminology used in the threat modeling domain. To be able to effectively utilize this artifact the practitioners need to be guided by someone with threat modeling expertise.

In the discussion session, a short cycle of the approach was completed. Three solutions were suggested to be feasible, based on the set of requirements that were collectively marked most important by the practitioners. At the end of the session, the results were presented to the practitioners, and it was explained they should look at them individually to select a final alternative. However, a few days after the session it was informally communicated that the outcome of the approach was not understood. The results indicate that our approach has potential. However, the execution of the approach needs to be refined before it can be effectively used in practice.

9.3 Post case survey results

The survey provided a mix of positive and negative statements to test the following criteria from the Prat taxonomy [137]: Effectiveness, accuracy, usefulness, ease of use, completeness, and functionality. Each statement was rated by the individual participants on a 5-point Likert scale, ranging from strongly agree to disagree and the middle was "not certain". Note, due to the limited number of participants, it was not possible to obtain conclusions with high certainty, therefore some potential conflicts and indications are discussed. The proposed approach was interpreted as slightly more effective for selecting a technique than a tool. The accuracy of the approach was leaning towards being slightly accurate. Although it was agreed that the method made the selection more structured and provided an effective solution, the usefulness in terms of simplicity was not agreed on. However, the participants suggested that they would be able to apply the method outside the context of the study. The completeness of the method was indicated to be lacking, the participants indicated they were not able to express their needs through the proposed prioritization. Lastly, the results regarding the functionality of the DSS were also inconclusive. There was a slight need for background knowledge. The perceived difficulty in using the DSS was contradictory.

9.4 Lessons learned

The validation case study indicated that there is still a need for improvements to our systematic selection approach. This section presents the lessons learned from the case study that can serve as a foundation for future research. During this study, the requirements have been improved in several iterations. However, the final set of requirements is still found to be multi-interpretable. The goal of making an approach that could be applied in multiple organisations led to the creation of abstract requirements. Because of the abstraction, each of these requirements has multiple implementations. When practitioners select a threat modeling method, they have some mental models of specific implementations they desire and specific implementations they want to avoid. Translating these specific implementations to more abstract requirements is a challenging task.

This challenge could be addressed in two ways. The requirements extracted so far can be split up into more specific sub-requirements. This solution is the least sought-after approach since it will increase the number of requirements significantly. The existing set of requirements has already been indicated as comprehensive due to the sheer number of requirements to prioritize. Increasing the number of requirements is expected to have a negative impact on the adaptation of our approach. The other option is to make some small adjustments to the terminology used and involve a domain expert who is experienced in threat modeling and requirement engineering. This domain expert can then assist with matching the specific mental models to the abstract requirements. When a group decision is made, the domain expert should already be actively involved in the preparation by providing assistance. This establishes a common ground for agreement on interpretations, aiming to facilitate a more streamlined and focused discussion session.

Additionally, the preliminary statements and questions presented to guide the practitioner through the prioritization of requirements were found to be multi-interpretable. To improve them, statements should be simplified and made modular. For example, there was a discussion about the statement *"I want to consider tool features and automation"*. During the case study, the participant's attention was primarily directed toward the automation aspect. It was not entirely evident if they intended to take tool features into account. Even when specifically inquired about tool features, the focus shifted back to automation. This resulted in a drawn-out discussion that could be resolved by having those components split up.

In addition to refining the content, certain aspects of the process were found to be sub-optimal. Because of the group decision format, each individual decision-maker was tasked with going through the preliminary questions and prioritization beforehand. The participants explained that there were still some unresolved questions about the contents, which were attempted to be addressed during the discussion session. However, not only was the number of questions too large to discuss in the allotted time, but not addressing these questions beforehand led to some assumptions made that sometimes affected the whole prioritization. For other participants, this was a reason to not proceed any further with the preliminary task, leading to partial completion. Consequently, it affected the discussion session, leading to an insufficient amount of time to adequately address the questions and introduce the contents to the participants that prematurely quit the preliminary task.

To address this issue, we will introduce an individual iteration of the preliminary task. In which the approach facilitator discusses and resolves these questions so that every piece of necessary content is understood by the participants beforehand. Also, the discussion session should be timeboxed, meaning that depending on the allocated time and the prioritization, the facilitator should focus on the biggest inconsistencies and allocate less time when there is only a slight difference in prioritization (e.g., *Should have* vs *Could have*). Lastly, at the end of the session, there should be a short block in which the outcome of the DSS is discussed and explained. Rather than just having a monologue, the facilitator should include the participants in the conversation to test if the outcome is understood and more importantly, if they are aware that they now should look deeper into these options, to select the method of their choice.

10 Concluding the research

10.1 Discussion

10.1.1 Systematic literature review

Despite the fact that threat modeling has been a subject of dedicated research for over two decades, a common definition of the concept has yet to be adopted. This posed a challenge for the systematic literature review. To deal with this, multiple draft queries were formulated and tested based on the inclusion of some of the most insightful papers obtained from the initial exploration. After the final query was formed and executed as explained in Section 4.2.1, only a couple of papers were found suitable. This is why backward snowballing has been conducted. It can be argued that the papers extracted as a result have varying quality and applications beyond threat modeling.

Concurrently to extracting qualities and requirements, a qualitative judgment on the potential applicability to the threat modeling domain has been made. Rather than documenting all criteria, only the ones that were judged to be potentially applicable were extracted. Because a significant number of qualities and requirements were gathered in the context of the intertwined domains of risk assessment and security requirement engineering, some were reformulated to be applicable in the context of threat modeling. This approach was taken due to the limited number of available papers that provide criteria for comparing threat modeling methods. Only considering dedicated threat modeling papers, would lead to a biased overview originating from less than a handful of papers. It can be argued that the qualities and requirements are therefore not a direct representation of the criteria in threat modeling method literature. To address this, every step that has been taken to obtain the results is documented. Rather than checking cross-paper citations, we decided to validate these results with experts. The validation in terms of expert surveys and interviews, indicates that both the set of qualities and requirements are an accurate representation of threat modeling comparison criteria.

10.1.2 Survey and expert interviews

The survey used to collect information on requirements and qualities served to validate the quality structure and the refined set of requirements extracted from the literature. It can be argued that the sample size is limited. This is mitigated by the careful selection of participants based on their expertise in the field. The survey provided valuable insights into the relevancy and applicability while simultaneously creating an outline for the interviews. The survey was extensive, comprising numerous questions. Participants brought up that this was challenging. It is possible that at a certain point, the concentration is lost, and participants perform a less thorough evaluation of the final few statements. It was observed that the comments in the first few parts were more elaborate. When the survey progressed the rate at which comments were placed decreased and the comments became shorter, with some exceptions. It can be questioned if these types of open comment questions are sufficient for extracting additional context. However, this is compensated by the follow-up interviews, during which brief comments were discussed, allowing participants to elaborate on the context. We considered this approach adequate for the refinement and validation of the study's concepts.

The expert interviews were conducted with the same participants as the survey. These interviews played a valuable role in uncovering insights and practical perspectives on threat modeling method selection. Also, the requirements and criteria were further developed. An elaborative coding process has been established to analyze the interviews as structured as possible. However, coding and manual analysis are still qualitative and contain subjective interpretations. Although some statements have been cross-coded, there were not enough resources to do it for all data collected. To provide transparency, all pseudonymized transcripts are attached to this project.

Additionally, some decision-makers expressed that they would select a threat modeling method by comparing alternatives based on goal-related characteristics. This primarily concerned tool-oriented selection approaches. Other experts selected a method based on the familiarity or context of the users. This was often associated with their vision of the goals and scope of threat modeling. It is not certain if the proposed systematic approach is sufficiently matching all contexts. A variety of future case studies can help provide an answer to this limitation.

10.1.3 The systematic approach and case study

Due to limited resources a solitary case study was conducted within a single large organisation, involving a group of practitioners who prioritized and identified their feature requirements using the MoSCoW method. Only having conducted a single case study can lead to a biased evaluation. To increase the validation maturity, more case studies should be performed covering companies in diverse domains with varying numbers of employees. The limited number of participants that have partaken in the case study focused their selection on a specific component of the threat modeling scope. This was dealt with in the design of the systematic approach by prioritizing what groups of requirements they wanted to consider beforehand. Nevertheless, this resulted in some discussions about requirement interpretation, as their focus was centered on the currently implemented tool and its limitations.

Autonomously combining and translating the mental models of the practitioners to the abstract requirements is found to be a challenging task that still is unresolved. Therefore, an expert in both threat modeling and requirement engineering should assist with resolving the discussions surrounding multi-interpretable requirements. Due to this issue, some practitioners did not complete the preparation, which potentially contributed to some of the observed misalignments. To test the full potential of the approach the lessons learned, elaborated on in Section 9.4, need to be transformed into refinements and the improved approach needs to be tested in multiple environments.

During the sessions, the practitioners collectively agreed upon the prioritization of the requirements. Although it was stated in the introduction that prioritizing using hard constraints would lead to disregarding solutions that were not in line with these constraints, this was not clear at the beginning of the discussion session. This led to some confusion among the participants, they expressed a reluctance to exclude solutions based on these hard constraints. To address this issue, participants were explicitly requested to re-evaluate the requirements they prioritized with a hard constraint, considering if the aspects were important enough to lead to the rejection of a solution. After this, it was observed that the practitioners often leaned towards using soft constraints.

Although the outcome led to some feasible solutions, this was based on prioritizing the most important requirements (according to the practitioners), which was 27% of the total requirements. The scores among the feasible alternatives were closely matched. While testing the approach it was observed that the number of feasible solutions decreased fast once the number of hard requirements increased. Additionally, a higher number of prioritized requirements, in general, resulted in greater score differences between the alternatives, leading to a more distinct ranking. To address this, the practitioners were welcomed to prioritize beyond the highlighted requirements. However, the time-restricted session did not allow for the group prioritization of these requirements.

The score calculation was done based on a weighted sum, where the prioritizations are the weights. MoSCoW is limited to four options while this leads to simplicity, it also limits the expression of importance. It was observed that practitioners felt the need to have a bigger spectrum they could prioritize on. For example, two requirements can be both prioritized as a *Must have*, but one of them adds significantly more value to the organisation and is therefore found more important. This is not accounted for in the score calculation. Also, all requirements are benchmarked using Boolean or an extended Boolean value that accounts for partial inclusion. In practice, some requirements can be non-Boolean, due to multiple levels of implementation. For example, portability concerns exporting and importing system diagrams. In the lowest level of implementation, portability regards the ability to import and export project files specific to the threat modeling tool. While on a higher level, it can import and export a variety of formats which makes the transition to the tool easier. Both levels of implementation can have different priorities. However, this is not accounted for in the approach.

The alternatives considered are a limited set of combined method components. Although these are carefully selected, it is important to note that there are other alternatives out there. Moreover, the lack of sufficient threat modeling tools for business applications affects the decision model. Tools were evaluated based on their documentation. As shown in the results of the SLR, the availability of the documentation was limited. It is not certain that the documentation of the selected tools fully represents their capabilities, which in turn can impact the benchmarking and decision model.

The selection of threat modeling methods is complex due to the misalignment in the definition of its concepts, its associated goals, and foreseen scope. Identifying and prioritizing requirements related to a concept that is multi-interpretable is a challenging task and practitioners are often not aware of the potential of other alternatives. The decision model proposed in this paper attempts to broaden this vision while tailoring it toward the preferences of the decision-maker. Besides contributing to making more informed decisions, an overview is provided of all the potential threat modeling components, requirements, and qualities.

10.2 Threats to validity

Assessment of validity threats is performed to identify limitations. The process is done structurally using a validation checklist and is strongly related to the theory of Wohlin et al. [198]. The full process can be found in Appendix A. The assessment findings will be summarized in the following sections, and proposed mitigations are highlighted.

10.2.1 Conclusion validity

Conclusion validity concerns issues that affect the ability to draw a correct conclusion [198]. Due to the explorative nature of this study, no advanced statistical tests are utilized. Conclusions are drawn based on a combination of occurrence, and averages, but mainly supported by qualitative interpretations. Data gathering for the decision model is conducted by doing an SLR, expert surveys, and interviews. Although the sample size of the survey is relatively small, this is mitigated by the strict selection of expert participants and doing follow-up interviews to supplement the results with qualitative data. The number of interviews is statistically satisfactory [127].

Each measurement is carefully designed, and decisions are documented and explained. The requirements which are a core part of the study are refined over time but have been kept consistent during each research activity. Nevertheless, it is important to note that these are still perceived as multi-interpretable by the case study participants, which harms the overall reliability of the measurements. Additionally, this study is conducted in combination with an internship at the case study company. Although the researcher does not gain anything from reporting specific results, the data dredging bias is still proactively mitigated through the construction of the study. By looking for relationships between concepts and then building a treatment based on these concepts, it is less likely that there is a bias caused by data dredging.

The core of the treatment is adapted from a published, and peer-reviewed framework [51]. The quality criteria were not considered in the final-decision model, but overall, the outline is maintained. The treatment is equally applied to all participants. However, these participants are part of one group decision-making process. Additionally, during this case study, not all participants performed the preparation activity equally, which can also harm the reliability. Lastly, the subjects for the expert interview are scattered over multiple organisations and countries, which increases sample heterogeneity. However, they are approached through utilizing personal networks and snowballing, which may harm the heterogeneity. The case study was done with a group of decision-makers from the same organisation. This increases homogeneity, which affects the external validity, but does not harm conclusion validity. To make the treatment more reliable and draw a valid conclusion on its practical use, multiple future case studies should be conducted.

10.2.2 Internal validity

Internal validity verifies the causality of relations between concepts of study [198]. The instrumentation used in the survey was perceived as exhaustive. Especially, the last view sections in which requirements were validated might be less thoroughly examined. This was also indicated by the decrease in comments and was further expressed during the interviews. Furthermore, the concepts tested were found to be multi-interpretable. To deal with this, requirements have been refined in between each research activity. Nevertheless, the final version was still perceived as multi-interpretable in the case study. This can cause an instrumentation bias, which affects the internal validity.

The selection of expert participants for data collection is done through purposive sampling [48] and by means of snowballing. The experts are a mix of actively recruited participants as well as volunteers. These participants were recruited based on professional affiliation with threat modeling, and their general work experience. Both types of participants might be more motivated than the average decision-maker, which makes this group less representative of the whole population. The selection of case study participants is executed by means of convenience. Due to the lack of resources and connections, this bias is accepted.

Additionally, during the interviews criteria are extracted and validated. Although the participants' work experience and familiarity have been verified, the study did not differentiate between levels of experience with threat modeling method selection. Another potential threat originates from the asynchronous preparation as part of the case study. This provides the potential for opinion-affecting interaction between participants, which is enlarged given that they are from the same organisation. On the other hand, the goal of the discussion session is to remove inconsistencies among the prioritizations, which is based on opinion-affecting interaction. In the end, this is not regarded as a threat.

10.2.3 Construct validity

Construct validity considers the construction of the experiment regarding the relationship between theory and observation [198]. This thesis creates a systematic decision-making approach utilizing a decision model for the threat modeling method selection. The designed approach is evaluated through a case study. The conducted case study is limited to a single instance, which harms the generalizability of the findings. Due to resource constraints, performing multiple case studies is future work. This has been repeatedly mentioned throughout the report.

To protect this study from evaluation apprehension, individual pressure is relieved by explaining the focus of the study and making explicit that individual performance will not be assessed. The case study enrolls a post-task questionnaire based on a set of carefully selected evaluation criteria. Because the test and treatment are separated, it is not expected to have any interaction bias. However, it is possible that the perception of the treatment is influenced by history, through interaction during the internship.

The participants of the case study anticipated the outcome of the decision model to be a single tool-based solution. Additionally, it was expected that the model also determined if threat modeling was relevant given organisational security implementations. However, the outcome of the decision model is a ranked list of multiple threat modeling methods, consisting of a technique, notation, and tool. Relevancy was determined in terms of requirement inclusion and did not consider existing security implementations. This may have negatively influenced the perception of the outcome because the guessed hypothesis did not match the actual outcome. Furthermore, there was potential ambiguity throughout the interview component of the study. A non-deliberate expectancy bias could occur by asking leading questions. The prepared interview guide and survey were checked for such bias by multiple individuals. Despite the case study being guided by the researcher, it is not anticipated that the researcher's implicit outcome expectancy would have influenced the prioritization input of the participants.

10.2.4 External validity

External validity regards the generalizability of results [198]. The experiment is performed in a business setting, where threat modeling is in a state of infancy. Applying the study to a different setting where threat modeling is unknown or well-established might yield different results. For instance, it is expected that in a well-established setting, the current technique already maintains numerous instances and that proposing a change would affect other established practices. Given these observations, participants could have an enlarged positive bias for the already implemented approach.

Further threats are concerning the participants of both the expert interviews and case studies. These participants are from diverse organisations across the globe. On the contrary, the case study is done within one large organisation in the Netherlands. This large organisation has a group of dedicated security officers who can make group decisions. In smaller companies elsewhere there might be a single stakeholder with multiple roles perceiving the approach and its contents differently. Lastly, it is observed that the threat modeling domain is continuously evolving. The proposed decision model is only valid given a snapshot of time. When repeating this study in the future, an updated decision model must be established. These threats are accepted due to the lack of resources. Throughout this thesis, limitations are explicitly reported to guide future research on interpreting the results.

10.3 Conclusion

This thesis contains an experimental approach for the systematical selection of a threat modeling method. For the core of the approach, we modeled the selection as a complex multi-criteria decision-making problem. This allowed for the evaluation of threat modeling methods based on a set of comparison criteria. We adapted the framework of [51] to create a decision model, based on weighted sum. A threat modeling method was split up into three core components: a technique, tool, and notation. The combination of these components represented a single alternative method. To achieve the objective of creating a systematic approach, some research questions were formulated and answered:

SRQ1: "How can existing threat modeling approaches be compared?"

To answer this question an SLR is conducted with a total harvest of 127 papers whereof 38 contained a comparison. The literature suggests there is no standardized approach for comparing threat modeling or related methods. Only a very limited number of papers were identified that specifically compare threat modeling methods or components of methods, which indicates a gap. By looking into related domains such as risk assessment and security requirement engineering, three different comparison techniques were identified. The most common one is a structured comparison based on a variety of criteria, ranging from open-ended textual to Boolean assessments. Other studies compare based on qualitative strengths and weaknesses or compare based on a summary of the methods. The number of tool comparisons found was limited and restricted by their narrow scope.

SRQ2: "What are the criteria for selecting threat modeling approaches?"

Criteria for selecting threat modeling approaches have been extracted and iteratively refined over multiple scientific methods. First, the results of the SLR identified 32 papers with criteria that could be potentially applied to the selection problem. After a refinement session with multiple researchers, this resulted in 41 requirements and 27 qualities. The requirements were reformulated and restructured into individual user stories. Both requirements and qualities were subjected to an expert survey. The results of the survey provide information on the relevancy of main requirements, the applicability of sub-user stories, and additional requirements and qualities. Furthermore, the relative quality importance was measured. The results of the survey were further investigated in a follow-up interview, which provided context to the selection criteria. Also, by exploring the current selection, additional requirements and qualities from practice were identified. Moreover, by doing an occurrence analysis the most mentioned criteria were identified. Based on the qualitative observations and results of these research steps, the requirements were refined once more, using terminology as close as possible to what was used by the experts, and the set was removed and restructured. This resulted in 108 modular requirements, of which 95 were used in the score calculation of the decision model, 21 quality criteria, and thirteen quality sub-criteria for the selection of threat modeling approaches.

> **SRQ3:** "What type of systematic approach can be used for threat modeling method selection?"

After comparing different categories of systematic decision-making approaches, it was proposed to use MCDM. Furthermore, looking into the different MCDM frameworks, it was concluded that the framework of Farshidi [51], had the most benefits given the context of the problem. This framework was simplified by removing the quality dimension, due to resource constraints. Half of each interview was dedicated to exploring the contexts of the threat modeling selection domain. Qualitative observations were reported regarding the current selection approach, the tool and technique paradigm, as well as the goal and scope of threat modeling. These observations served as input for designing the systematic approach that is centered around the decision model. An explorative global multiplier was utilized to deal with non-functional requirements, that necessitate context beyond the method.

The decision model was developed by using a subset of the SLR results. The current version consists of eighteen threat modeling methods which are constructed by combining modular techniques, and commonly available tools, and subsequently matching the modeling notation to these components. Also, 95 requirements are considered as part of the score calculation. An assessment of relevancy has been made in which each requirement is rated based on what component(s) can be used to benchmark it. Furthermore, each component was individually benchmarked given the relevant requirement subset. Partial supportability was assigned in cases where a requirement was not explicitly supported by the component but showed partial dependency on the surrounding process. Finally, the benchmarks were combined to construct the final decision model.

SRQ4: "How well is the proposed decision-making method perceived by the stakeholders?"

To test the proposed decision-making method, a case study was performed within a large company. The designed approach has been enrolled for a group of information security officers. The post-task questionnaire contained a mix of positive and negative statements to test the effectiveness, accuracy, usefulness, ease of use, completeness, and functionality. Due to the limited number of participants, it was not possible to obtain conclusions with high certainty. Therefore, potential conflicts and indications were discussed. It was agreed that the method made the selection more structured and provided an effective solution. However, participants indicated that the contents of the approach were multi-interpretable, and they were not able to express all their needs through the proposed prioritization. The results indicate that the method itself has some potential. However, the approach needs to be refined before it can be effectively used in practice. Lessons learned were discussed and refinements to both the contents as well as the process are suggested for future work.

Based on the findings obtained from the sub-research question, we can answer the main research question:

MRQ:"How can organisations systematically select a threat modeling method?"

The proposed systematic selection approach is indicated to have potential for assisting in making traceable decisions but needs to be further validated in future research. As suggested by other research, there is no common definition for threat modeling in literature [200]. This study affirms that this translates into practice in terms of a lack of a common alignment in preference, goals, and scope. These factors are observed to influence the selection of threat modeling methods. The varying mental models make it very challenging to systematically select a method based on a set of common requirements. However, the extensive range of collected data, and particularly the methods, requirements, and quality criteria refined through a multi-phased research design, can serve as a foundation for future research.

10.4 Future work

Future work is encouraged to further validate the approach by testing it in different environments. It is expected to be beneficial to test the approach in smaller organisations, those where decision-making is influenced by a single individual, and organisations located outside of the Netherlands. Additionally, when proceeding with this, it is recommended to especially account for the threat modeling experience of the decision-makers that partake in the case studies. Furthermore, future work is invited to refine the current approach based on the lessons learned and new observations made.

Additionally, a formal investigation could be conducted on how organisational context, such as the security maturity and the current process of doing threat modeling, affects the preferences of the decision-makers. This can contribute towards a greater understanding of how decisions in cyber security are made and how this should be handled in a systematic approach. Moreover, the quality x requirement mapping could be refined to validate the hypothesized relationships established based on the co-occurrence analysis. When motivated, it can even be extended by considering multiple types of relationships, such as relationships between qualities. Refining the mapping enables the incorporation of the quality dimension into the decision model.

Research could also explore the non-functional requirements that require additional context beyond the method itself to test their satisfaction. In this study, an explorative global multiplier is proposed to deal with them, which results in an additional score for each alternative. However, the weights are experimental and only familiarity is accounted for. Furthermore, other non-Boolean requirements could be considered as an extension of the decision model. Additional relative weights can be added to deal with ties between alternatives or to take into account the difference in importance between equally prioritized requirements. Moreover, the decision model can be extended by adding more combinations of techniques, tools of methods. Complex methods that consist of multiple techniques could be considered but should be dealt with accordingly in the decision model. In addition, automated threat modeling approaches could be included. Future work should decide whether the existing model adequately covers these approaches or if a new model, encompassing distinct requirements, needs to be developed.

To integrate threat modeling into the development process, research is needed to explore the role of risk in threat modeling. The coded transcripts contain segments that have been designated with a memo, which can serve as a starting point for exploring this perceived relationship. Exploring this relationship can have a great contribution towards not only streamlining threat modeling but also determining the added value of threat modeling methods in comparison to other related security activities. This type of added value is suggested by some as being the core contributor to usefulness, the most mentioned quality criterion throughout all the interviews.

10.5 Data repository

The de-identified interview transcripts and survey results generated in this research project can be requested by email from the supervisor, Kate Labunets. At the thesis publication date, her email is k.labunets@uu.nl.

Bibliography

[1] Abe, T., Hayashi, S., Saeki, M.: Modeling security threat patterns to derive negative scenarios. In: 2013 20th Asia-Pacific Software Engineering Conference (APSEC). vol. 1, pp. 58–66. IEEE (2013)

[2] Alberts, C., Dorofee, A., Stevens, J., Woody, C.: Introduction to the octave approach. Pittsburgh, PA, Carnegie Mellon University pp. 72–74 (2003)

[3] Almorsy, M., Grundy, J., Ibrahim, A.S.: Automated software architecture security risk analysis using formalized signatures. In: 2013 35th International Conference on Software Engineering (ICSE). pp. 662–671. IEEE (2013)

[4] Ansari, M.T.J., Pandey, D., Alenezi, M.: Store: Security threat oriented requirements engineering methodology. Journal of King Saud University-Computer and Information Sciences **34**(2), 191–203 (2022)

[5] Arsac, W., Bella, G., Chantry, X., Compagna, L.: Multi-attacker protocol validation. Journal of Automated Reasoning **46**(3-4), 353–388 (2011)

[6] Aruldoss, M., Lakshmi, T.M., Venkatesan, V.P.: A survey on multi criteria decision making methods and its applications. American Journal of Information Systems **1**(1), 31–43 (2013)

[7] Aven, T.: Risk analysis. John Wiley & Sons (2015)

[8] Beckers, K., Côté, I., Faßbender, S., Heisel, M., Hofbauer, S.: A pattern-based method for establishing a cloud-specific information security management system: Establishing information security management systems for clouds considering security, privacy, and legal compliance. Requirements Engineering **18**, 343–395 (2013)

[9] Beckers, K., Hatebur, D., Heisel, M.: A problem-based threat analysis in compliance with common criteria. In: 2013 International Conference on Availability, Reliability and Security. pp. 111–120. IEEE (2013)

[10] Bedi, P., Gandotra, V., Singhal, A., Narang, H., Sharma, S.: Threat-oriented security framework in risk management using multiagent system. Software: Practice and Experience **43**(9), 1013–1038 (2013)

[11] Berger, B.J., Sohr, K., Koschke, R.: Automatically extracting threats from extended data flow diagrams. In: Engineering Secure Software and Systems: 8th International Symposium, ESSoS 2016, London, UK, April 6–8, 2016. Proceedings 8. pp. 56–71. Springer (2016)

[12] Van den Berghe, A., Scandariato, R., Yskout, K., Joosen, W.: Design notations for secure software: a systematic literature review. Software & Systems Modeling **16**, 809–831 (2017)

[13] Bhatti, B.M., Mubarak, S., Nagalingam, S.: Information security risk management in it outsourcing–a quarter-century systematic literature review. Journal of Global Information Technology Management **24**(4), 259–298 (2021)

[14] Bilbao-Terol, A., Arenas-Parra, M., Cañal-Fernández, V., Antomil-Ibias, J.: Using topsis for assessing the sustainability of government bond funds. Omega **49**, 1–17 (2014)

[15] Bishop, M.: What is computer security? IEEE Security & Privacy **1**(1), 67–69 (2003)

[16] Brændeland, G., Refsdal, A., Stølen, K.: Modular analysis and modelling of risk scenarios with dependencies. Journal of Systems and Software **83**(10), 1995–2013 (2010)

[17] Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and software technology **38**(4), 275–280 (1996)

[18] Brinkkemper, S., Saeki, M., Harmsen, F.: Assembly techniques for method engineering. In: Advanced Information Systems Engineering: 10th International Conference, CAiSE'98 Pisa, Italy, June 8–12, 1998 Proceedings 10. pp. 381–400. Springer (1998)

[19] Brinkkemper, S., Saeki, M., Harmsen, F.: Meta-modelling based assembly techniques for situational method engineering. Information Systems **24**(3), 209–228 (1999)

[20] Burmester, M., Magkos, E., Chrissikopoulos, V.: Modeling security in cyber–physical systems. International journal of critical infrastructure protection **5**(3-4), 118–126 (2012)

[21] Büyüközkan, G., Kahraman, C., Ruan, D.: A fuzzy multi-criteria decision approach for software development strategy selection. International journal of general systems **33**(2-3), 259–280 (2004)

[22] Bygdås, E., Jaatun, L.A., Antonsen, S.B., Ringen, A., Eiring, E.: Evaluating threat modeling tools: Microsoft tmt versus owasp threat dragon. In: 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA). pp. 1–7. IEEE (2021)

[23] Caralli, R.A., Stevens, J.F., Young, L.R., Wilson, W.R.: Introducing octave allegro: Improving the information security risk assessment process. Hansom AFB, MA (2007)

[24] Casola, V., De Benedictis, A., Rak, M., Villano, U.: Toward the automation of threat modeling and risk assessment in iot systems. Internet of Things **7**, 100056 (2019)

[25] Cerotti, D., Codetta-Raiteri, D., Egidi, L., Franceschinis, G., Portinale, L., Dondossola, G., Terruggia, R.: Analysis and detection of cyber attack processes targeting smart grids. In: 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe). pp. 1–5. IEEE (2019)

[26] Cerotti, D., Codetta-Raiteri, D., Dondossola, G., Egidi, L., Franceschinis, G., Portinale, L., Terruggia, R.: Evidence-based analysis of cyber attacks to security monitored distributed energy resources. Applied Sciences **10**(14), 4725 (2020)

[27] Chai, J., Liu, J.N., Ngai, E.W.: Application of decision-making techniques in supplier selection: A systematic review of literature. Expert systems with applications **40**(10), 3872–3885 (2013)

[28] Chen, Y., Boehm, B., Sheppard, L.: Value driven security threat modeling based on attack path analysis. In: 2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07). pp. 280a–280a. IEEE (2007)

[29] Cherdantseva, Y., Burnap, P., Blyth, A., Eden, P., Jones, K., Soulsby, H., Stoddart, K.: A review of cyber security risk assessment methods for scada systems. Computers & security **56**, 1–27 (2016)

[30] Cleland-Huang, J.: How well do you know your personae non gratae? IEEE software **31**(4), 28–31 (2014)

[31] Consortium, A.B.: Chapter 10: Moscow prioritisation (Jan 2014), https://www.agilebusiness.org/dsdm-project-framework/moscow-prioririsation.html

[32] Consortium, A.B., et al.: The DSDM Agile Project Framework. DSDM Consortium (2014)

[33] Crozier, R., Ranyard, R.: Cognitive process models and explanations of decision making. In: Decision making, pp. 19–34. Routledge (2002)

[34] Daramola, O., Pan, Y., Karpati, P., Sindre, G.: A comparative review of istar-based and use case-based security modelling initiatives. In: 2012 Sixth International Conference on Research Challenges in Information Science (RCIS). pp. 1–12. IEEE (2012)

[35] De Keyser, W.S., Peeters, P.: Argus—a new multiple criteria method based on the general idea of outranking. In: Applying multiple criteria aid for decision to environmental management, pp. 263–278. Springer (1994)

[36] Demirtas, E.A., Ustun, O.: Analytic network process and multi-period goal programming integration in purchasing decisions. Computers & Industrial Engineering **56**(2), 677–690 (2009)

[37] Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering **16**(1), 3–32 (2011)

[38] Dhillon, D.: Developer-driven threat modeling: Lessons learned in the trenches. IEEE Security & Privacy **9**(4), 41–47 (2011)

[39] Dimitrakos, T., Ritchie, B., Raptis, D., Stølen, K.: Model based security risk analysis for web applications: the coras approach. In: EuroWeb 2002 Conference. pp. 1–13 (2002)

[40] Dodgson, J.S., Spackman, M., Pearman, A., Phillips, L.D.: Multi-criteria analysis: a manual (2009)

[41] Dvořák, O., Pergl, R., Kroha, P.: Affordance-driven software assembling. In: Advances in Enterprise Engineering XII: 8th Enterprise Engineering Working Conference, EEWC 2018, Luxembourg, Luxembourg, May 28–June 1, 2018, Proceedings. pp. 39–54. Springer (2018)

[42] Ekstedt, M., Johnson, P., Lagerström, R., Gorton, D., Nydrén, J., Shahzad, K.: Securi cad by foreseeti: A cad tool for enterprise cyber security management. In: 2015 IEEE 19th international enterprise distributed object computing workshop. pp. 152–155. IEEE (2015)

[43] El Ariss, O., Xu, D.: Modeling security attacks with statecharts. In: Proceedings of the joint ACM SIGSOFT conference–QoSA and ACM SIGSOFT symposium–ISARCS on Quality of software architectures–QoSA and architecting critical systems–ISARCS. pp. 123–132 (2011)

[44] Elahi, G., Yu, E.: A goal oriented approach for modeling and analyzing security trade-offs. In: Conceptual Modeling-ER 2007: 26th International Conference on Conceptual Modeling, Auckland, New Zealand, November 5-9, 2007. Proceedings 26. pp. 375–390. Springer (2007)

[45] Elahi, G., Yu, E., Zannone, N.: A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. Requirements engineering **15**, 41–62 (2010)

[46] Elmrabit, N., Yang, S.H., Yang, L., Zhou, H.: Insider threat risk prediction based on bayesian network. Computers & Security **96**, 101908 (2020)

[47] Encina, C.O., Fernandez, E.B., Monge, A.R.: Threat analysis and misuse patterns of federated inter-cloud systems. In: Proceedings of the 19th European Conference on Pattern Languages of Programs. pp. 1–8 (2014)

[48] Etikan, I., Musa, S.A., Alkassim, R.S., et al.: Comparison of convenience sampling and purposive sampling. American journal of theoretical and applied statistics **5**(1), 1–4 (2016)

[49] Fabian, B., Gürses, S., Heisel, M., Santen, T., Schmidt, H.: A comparison of security requirements engineering methods. Requirements engineering **15**, 7–40 (2010)

[50] Farah, A., Saida, B., Mourad, O.C.: On the security of business processes: classification of approaches, comparison, and research directions. In: 2021 International Conference on Networking and Advanced Systems (ICNAS). pp. 1–8. IEEE (2021)

[51] Farshidi, S.: Multi-Criteria Decision-Making in Software Production. Ph.D. thesis, Utrecht University (2020)

[52] Farshidi, S., Jansen, S., España, S., Verkleij, J.: Decision support for blockchain platform selection: Three industry case studies. IEEE Transactions on Engineering Management **67**(4), 1109–1128 (2020)

[53] Fishburn, P.C.: Exceptional paper—lexicographic orders, utilities and decision rules: A survey. Management science **20**(11), 1442–1471 (1974)

[54] Flechais, I., Mascolo, C., Sasse, M.A.: Integrating security and usability into the requirements and design process. International Journal of Electronic Security and Digital Forensics **1**(1), 12–26 (2007)

[55] Floudas, C.A., Pardalos, P.M.: Encyclopedia of optimization. Springer Science & Business Media (2008)

[56] Frydman, M., Ruiz, G., Heymann, E., César, E., Miller, B.P., et al.: Automating risk analysis of software design models. The Scientific World Journal **2014** (2014)

[57] Geismann, J., Bodden, E.: A systematic literature review of model-driven security engineering for cyber–physical systems. Journal of Systems and Software **169**, 110697 (2020)

[58] Gholami, A., Laure, E.: Advanced cloud privacy threat modeling. arXiv preprint arXiv:1601.01500 (2016)

[59] Granata, D., Rak, M.: Design and development of a technique for the automation of the risk analysis process in it security. In: CLOSER. pp. 87–98 (2021)

[60] Granata, D., Rak, M., Salzillo, G.: Automated threat modeling approaches: Comparison of open source tools. In: International Conference on the Quality of Information and Communications Technology. pp. 250–265. Springer (2022)

[61] Green, P.E., Krieger, A.M., Agarwal, M.K.: A cross validation test of four models for quantifying multiattribute preferences. Marketing Letters **4**(4), 369–380 (1993)

[62] Güneri, A.F., Ertay, T., Yücel, A.: An approach based on anfis input selection and modeling for supplier selection problem. Expert Systems with Applications **38**(12), 14907–14917 (2011)

[63] Hagen, E., Reksten, E., et al.: Seamonster - security modeling software files, `https://sourceforge.net/projects/seamonster/files/SeaMonster%205/`

[64] Haley, C., Laney, R., Moffett, J., Nuseibeh, B.: Security requirements engineering: A framework for representation and analysis. IEEE Transactions on Software Engineering **34**(1), 133–153 (2008)

[65] Haley, C.B., Moffett, J.D., Laney, R., Nuseibeh, B.: A framework for security requirements engineering. In: Proceedings of the 2006 international workshop on Software engineering for secure systems. pp. 35–42 (2006)

[66] Halkidis, S.T., Tsantalis, N., Chatzigeorgiou, A., Stephanides, G.: Architectural risk analysis of software systems based on security patterns. IEEE Transactions on Dependable and Secure Computing **5**(3), 129–142 (2008)

[67] Hanine, M., Boutkhoum, O., Tikniouine, A., Agouti, T.: Application of an integrated multi-criteria decision making ahp-topsis methodology for etl software selection. SpringerPlus **5**(1), 1–17 (2016)

[68] Harmsen, A.F., Brinkkemper, J.N., Oei, J.H.: Situational method engineering for information system project approaches. University of Twente, Department of Computer Science (1994)

[69] Harris, R.: Introduction to decision making, virtualsalt. Online http://www. virtualsalt. com/crebook5. htm (accessed on 09/10/2011) (1998)

[70] Hassan, R., Bohner, S., El-Kassas, S., Eltoweissy, M.: Goal-oriented, b-based formal derivation of security design specifications from security requirements. In: 2008 Third International Conference on Availability, Reliability and Security. pp. 1443–1450. IEEE (2008)

[71] Hatebur, D., Heisel, M.: Problem frames and architectures for security problems. In: Computer Safety, Reliability, and Security: 24th International Conference, SAFECOMP 2005, Fredrikstad, Norway, September 28-30, 2005. Proceedings 24. pp. 390–404. Springer (2005)

[72] He, W., Zhang, Z.J., Li, W.: Information technology solutions, challenges, and suggestions for tackling the covid-19 pandemic. International journal of information management **57**, 102287 (2021)

[73] Hinloopen, E., Nijkamp, P., Rietveld, P.: Qualitative discrete multiple criteria choice models in regional planning. Regional Science and Urban Economics **13**(1), 77–102 (1983)

[74] Holm, H., Shahzad, K., Buschle, M., Ekstedt, M.: P2cysemol: Predictive, probabilistic cyber security modeling language. IEEE Transactions on Dependable and Secure Computing **12**(6), 626–639 (2014)

[75] Huang, Y.L., Cárdenas, A.A., Amin, S., Lin, Z.S., Tsai, H.Y., Sastry, S.: Understanding the physical and economic consequences of attacks on control systems. International Journal of Critical Infrastructure Protection **2**(3), 73–83 (2009)

[76] Hussain, S., Rasool, G., Atef, M., Shahid, A.K.: A review of approaches to model security into software systems. Journal of Basic and Applied Scientific Research **3**(4), 642–647 (2013)

[77] Ishizaka, A., Nemery, P.: Multi-criteria decision analysis: methods and software. John Wiley & Sons (2013)

[78] Istarwiki.org: I* wiki | i* tools, http://istar.rwth-aachen.de/tiki-index.php?page=i%2A+Tools

[79] Johnson, P., Vernotte, A., Ekstedt, M., Lagerström, R.: pwnpr3d: an attack-graph-driven probabilistic threat-modeling approach. In: 2016 11th international conference on availability, reliability and security (ARES). pp. 278–283. IEEE (2016)

[80] Johnson, P., Vernotte, A., Gorton, D., Ekstedt, M., Lagerström, R.: Quantitative information security risk estimation using probabilistic attack graphs. In: Risk Assessment and Risk-Driven Quality Assurance: 4th International Workshop, RISK 2016, Held in Conjunction with ICTSS 2016, Graz, Austria, October 18, 2016, Revised Selected Papers 4. pp. 37–52. Springer (2017)

[81] Jouini, M., Rabai, L.B.A., Aissa, A.B.: Classification of security threats in information systems. Procedia Computer Science **32**, 489–496 (2014)

[82] Jürjens, J., Shabalin, P.: Tools for secure systems development with uml. International Journal on Software Tools for Technology Transfer **9**(5-6), 527–544 (2007)

[83] Kammüller, F., Nurse, J.R., Probst, C.W.: Attack tree analysis for insider threats on the iot using isabelle. In: Human Aspects of Information Security, Privacy, and Trust: 4th International Conference, HAS 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings 4. pp. 234–246. Springer (2016)

[84] Kammüller, F., Probst, C.W.: Modeling and verification of insider threats using logical analysis. IEEE systems journal **11**(2), 534–545 (2015)

[85] Kamongi, P., Gomathisankaran, M., Kavi, K.: Nemesis: Automated architecture for threat modeling and risk assessment for cloud computing. In: Proc. 6th ASE International Conference on Privacy, Security, Risk and Trust (PASSAT) (2014)

[86] Karpati, P., Opdahl, A.L., Sindre, G.: Harm: Hacker attack representation method. In: Software and Data Technologies: 5th International Conference, ICSOFT 2010, Athens, Greece, July 22-24, 2010. Revised Selected Papers 5. pp. 156–175. Springer (2013)

[87] Karpati, P., Sindre, G., Opdahl, A.L.: Visualizing cyber attacks with misuse case maps. In: Requirements Engineering: Foundation for Software Quality: 16th International Working Conference, REFSQ 2010, Essen, Germany, June 30–July 2, 2010. Proceedings 16. pp. 262–275. Springer (2010)

[88] Khan, R.A., Khan, S.U., Alzahrani, M., Ilyas, M.: Security assurance model of software development for global software development vendors. Ieee Access **10**, 58458–58487 (2022)

[89] Khwaja, A.A., Urban, J.E.: A synthesis of evaluation criteria for software specifications and specification techniques. International Journal of Software Engineering and Knowledge Engineering **12**(05), 581–599 (2002)

[90] Kim, Y.G., Cha, S.: Threat scenario-based security risk analysis using use case modeling in information systems. Security and Communication Networks **5**(3), 293–300 (2012)

[91] Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University **33**(2004), 1–26 (2004)

[92] Kong, J., Xu, D., Zeng, X.: Uml-based modeling and analysis of security threats. International Journal of Software Engineering and Knowledge Engineering **20**(06), 875–897 (2010)

[93] Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Attack–defense trees. Journal of Logic and Computation **24**(1), 55–87 (2014)

[94] Kornyshova, E., Salinesi, C.: Mcdm techniques selection approaches: state of the art. In: 2007 ieee symposium on computational intelligence in multi-criteria decision-making. pp. 22–29. IEEE (2007)

[95] Kou, G., Ergu, D., Lin, C., Chen, Y.: Pairwise comparison matrix in multiple criteria decision making. Technological and economic development of economy **22**(5), 738–765 (2016)

[96] Kumar, A., Sah, B., Singh, A.R., Deng, Y., He, X., Kumar, P., Bansal, R.: A review of multi criteria decision making (mcdm) towards sustainable renewable energy development. Renewable and Sustainable Energy Reviews **69**, 596–609 (2017)

[97] Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) Proc. 23rd International Conference on Computer Aided Verification (CAV'11). LNCS, vol. 6806, pp. 585–591. Springer (2011)

[98] Lagerström, R., Xiong, W., Ekstedt, M.: Threat modeling and attack simulations of smart cities: A literature review and explorative study. ICISSP pp. 369–376 (2020)

[99] Larcom, B., Smith, S., Saitta, E.: Microsoft threat modeling tool, http://www.octotrike.org/

[100] Li, T., Paja, E., Mylopoulos, J., Horkoff, J., Beckers, K.: Security attack analysis using attack patterns. In: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS). pp. 1–13. IEEE (2016)

[101] Limited.., A.T.: Attack tree-based threat modeling – know how they will attack!, https://www.amenaza.com/documents.php

[102] Liu, L., Yu, E., Mylopoulos, J.: Security and privacy requirements analysis within a social setting. In: Proceedings. 11th IEEE International Requirements Engineering Conference, 2003. pp. 151–161. IEEE (2003)

[103] Lucio, L., Zhang, Q., Nguyen, P.H., Amrani, M., Klein, J., Vangheluwe, H., Le Traon, Y.: Advances in model-driven security. In: Advances in Computers, vol. 93, pp. 103–152. Elsevier (2014)

[104] Lund, M.S., Solhaug, B., Stølen, K.: Model-driven risk analysis: the CORAS approach. Springer Science & Business Media (2010)

[105] Lund, M.S., Solhaug, B., Stølen, K., Lund, M.S., Solhaug, B., Stølen, K.: A guided tour of the coras method. Model-driven risk analysis: The CORAS approach pp. 23–43 (2011)

[106] MAÊDA, N., RODRIGUES, M.V.G., Ângelo, M., MOREIRA, L., GOMES, C.F.S., d dos SANTOS, M.: Algorithm selection for machine learning classification: an application of the melchior multicriteria method. Modern Management Based on Big Data II and Machine Learning and Intelligent Systems III: Proceedings of MMBD 2021 and MLIS 2021 **341**, 154 (2021)

[107] Maheshwari, V., Prasanna, M.: Integrating risk assessment and threat modeling within sdlc process. In: 2016 international conference on inventive computation technologies (ICICT). vol. 1, pp. 1–5. IEEE (2016)

[108] Majumder, M., Majumder, M.: Multi criteria decision making. Impact of urbanization on water shortage in face of climatic aberrations pp. 35–47 (2015)

[109] Manzoor, S., Zhang, H., Suri, N.: Threat modeling and analysis for the cloud ecosystem. In: 2018 IEEE International Conference on Cloud Engineering (IC2E). pp. 278–281. IEEE (2018)

[110] Marle, F., Gidel, T.: A multi-criteria decision-making process for project risk management method selection. International Journal of Multicriteria Decision Making **2**(2), 189–223 (2012)

[111] Marle, F., Gidel, T.: Assisting project risk management method selection. International Journal of Project Organisation and Management **6**(3), 254–282 (2014)

[112] Marler, R.T., Arora, J.S.: The weighted sum method for multi-objective optimization: new insights. Structural and multidisciplinary optimization **41**, 853–862 (2010)

[113] Mauw, S., Oostdijk, M.: Foundations of attack trees. In: Information Security and Cryptology-ICISC 2005: 8th International Conference, Seoul, Korea, December 1-2, 2005, Revised Selected Papers 8. pp. 186–198. Springer (2006)

[114] Mavropoulos, O., Mouratidis, H., Fish, A., Panaousis, E.: Asto: A tool for security analysis of iot systems. In: 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA). pp. 395–400. IEEE (2017)

[115] McDermott, J., Fox, C.: Using abuse case models for security requirements analysis. In: Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99). pp. 55–64. IEEE (1999)

[116] Mead, N.R., Shull, F., Vemuru, K., Villadsen, O.: A hybrid threat modeling method. Carnegie MellonUniversity-Software Engineering Institute-Technical Report-CMU/SEI-2018-TN-002 (2018)

[117] Mellado, D., Blanco, C., Sánchez, L.E., Fernández-Medina, E.: A systematic review of security requirements engineering. Computer Standards & Interfaces **32**(4), 153–165 (2010)

[118] Mercuri, R.T.: Analyzing security costs. Communications of the ACM **46**(6), 15–18 (2003)

[119] Microsoft: Microsoft threat modeling tool, `https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool`

[120] Miranda, E.: Moscow rules: A quantitative exposé. In: Stray, V., Stol, K.J., Paasivaara, M., Kruchten, P. (eds.) Agile Processes in Software Engineering and Extreme Programming. pp. 19–34. Springer International Publishing, Cham (2022)

[121] Misra, S.K., Ray, A.: Comparative study on different multi-criteria decision making tools in software project selection scenario. International Journal of Advanced Research in Computer Science **3**(4) (2012)

[122] Mohsin, M., Sardar, M.U., Hasan, O., Anwar, Z.: Iotriskanalyzer: A probabilistic model checking based framework for formal risk analytics of the internet of things. IEEE Access **5**, 5494–5505 (2017)

[123] Mouratidis, H., Giorgini, P.: Secure tropos: a security-oriented extension of the tropos methodology. International Journal of Software Engineering and Knowledge Engineering **17**(02), 285–309 (2007)

[124] Mouratidis, H., Giorgini, P., Manson, G.: Modelling secure multiagent systems. In: Proceedings of the second international joint conference on Autonomous agents and multiagent systems. pp. 859–866 (2003)

[125] Muñante, D., Chiprianov, V., Gallon, L., Aniorté, P.: A review of security requirements engineering methods with respect to risk analysis and model-driven engineering. In: Availability, Reliability, and Security in Information Systems: IFIP WG 8.4, 8.9, TC 5 International Cross-Domain Conference, CD-ARES 2014 and 4th International Workshop on Security and Cognitive Informatics for Homeland Defense, SeCIHD 2014, Fribourg, Switzerland, September 8-12, 2014. Proceedings 9. pp. 79–93. Springer (2014)

[126] Musa Shuaibu, B., Md Norwawi, N., Selamat, M.H., Al-Alwani, A.: Systematic review of web application security development model. Artificial Intelligence Review **43**, 259–276 (2015)

[127] Namey, E., Guest, G., McKenna, K., Chen, M.: Evaluating bang for the buck: a cost-effectiveness comparison between individual interviews and focus groups based on thematic saturation levels. American Journal of Evaluation **37**(3), 425–440 (2016)

[128] National Institute of Standards and Technology: Guide for Conducting Risk, Assessments (2012)

[129] Neto, A.J.H., dos Santos, A.F.P.: Cyber threat hunting through automated hypothesis and multi-criteria decision making. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 1823–1830. IEEE (2020)

[130] Nguyen, P.H., Klein, J., Le Traon, Y., Kramer, M.E.: A systematic review of model-driven security. In: 2013 20th Asia-Pacific Software Engineering Conference (APSEC). vol. 1, pp. 432–441. IEEE (2013)

[131] OWASP: Owasp threat dragon, https://owasp.org/www-project-threat-dragon/docs-2/

[132] Oxford University Press: Oxford English Dictionary (2022)

[133] Perini, A., Ricca, F., Susi, A.: Tool-supported requirements prioritization: Comparing the ahp and cbrank methods. Information and Software Technology **51**(6), 1021–1032 (2009)

[134] Pfleeger, C.P.: Security in computing. Pearson Education India (2009)

[135] Pfleeger, C.P., Pfleeger, S.L.: Analyzing computer security: A threat/vulnerability/countermeasure approach. Prentice Hall Professional (2012)

[136] Potteiger, B., Martins, G., Koutsoukos, X.: Software and attack centric integrated threat modeling for quantitative risk assessment. In: Proceedings of the Symposium and Bootcamp on the Science of Security. pp. 99–108 (2016)

[137] Prat, N., Comyn-Wattiau, I., Akoka, J.: A taxonomy of evaluation methods for information systems artifacts. Journal of Management Information Systems **32**(3), 229–267 (2015)

[138] Rak, M., Salzillo, G., Granata, D.: Esseca: An automated expert system for threat modelling and penetration testing for iot ecosystems. Computers and Electrical Engineering **99**, 107721 (2022)

[139] Ramanathan, R., Ganesh, L.: Energy resource allocation incorporating qualitative and quantitative criteria: An integrated model using goal programming and ahp. Socio-Economic Planning Sciences **29**(3), 197–218 (1995)

[140] Raspotnig, C., Opdahl, A.: Comparing risk identification techniques for safety and security requirements. Journal of systems and software **86**(4), 1124–1151 (2013)

[141] Renatus, S., Teichmann, C., Eichler, J.: Method selection and tailoring for agile threat assessment and mitigation. In: 2015 10th International Conference on Availability, Reliability and Security. pp. 548–555. IEEE (2015)

[142] Reyes, J., Fuertes, W., Macas, M.: Development processes of vulnerability detection systems: A systematic review, approaches, challenges, and future directions. In: International Conference on Applied Technologies. pp. 335–350. Springer (2021)

[143] Rhee, K., Won, D., Jang, S.W., Chae, S., Park, S.: Threat modeling of a mobile device management system for secure smart work. Electronic Commerce Research **13**, 243–256 (2013)

[144] Ribeiro, R.A., Moreira, A.M., Van den Broek, P., Pimentel, A.: Hybrid assessment method for software engineering decisions. Decision Support Systems **51**(1), 208–219 (2011)

[145] Roubens, M.: Preference relations on actions and criteria in multicriteria decision making. European Journal of Operational Research **10**(1), 51–55 (1982)

[146] Saini, V., Duan, Q., Paruchuri, V.: Threat modeling using attack trees. Journal of Computing Sciences in Colleges **23**(4), 124–131 (2008)

[147] Saitta, P., Larcom, B., Eddington, M.: Trike v. 1 methodology document [draft]. URL: http://dymaxion. org/trike/Trike v1 Methodology Documentdraft. pdf (2005)

[148] Saldaña, J.: The coding manual for qualitative researchers. sage (2021)

[149] Salini, P., Kanmani, S.: Survey and analysis on security requirements engineering. Computers & Electrical Engineering **38**(6), 1785–1797 (2012)

[150] Scandariato, R., Wuyts, K., Joosen, W.: A descriptive study of microsoft's threat modeling technique. Requirements Engineering **20**, 163–180 (2015)

[151] Schaad, A.: Project ovvl–threat modeling support for the entire secure development lifecycle. SICHERHEIT 2020 (2020)

[152] Schaad, A., Borozdin, M.: Tam2: automated threat analysis. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp. 1103–1108 (2012)

[153] Schaad, A., Reski, T.: " open weakness and vulnerability modeler"(ovvl)–an updated approach to threat modeling. In: Proceedings of the 16th International Joint Conference on e-Business and Telecommunications, Prague, Czech Republic-Volume 2: SECRYPT. vol. 2, pp. 417–424 (2019)

[154] Schilling, A., Werners, B.: A quantitative threat modeling approach to maximize the return on security investment in cloud computing. In: Proceedings of the International Conference on Cloud Security Management ICCSM, Reading, UK. pp. 68–78 (2013)

[155] Schneier, B.: Attack trees. Dr. Dobb's journal **24**(12), 21–29 (1999)

[156] Selin, J.: Evaluation of threat modeling methodologies. Master's thesis, School of Technology (2019)

[157] Shamala, P., Ahmad, R., Yusoff, M.: A conceptual framework of info structure for information security risk assessment (isra). Journal of Information Security and Applications **18**(1), 45–52 (2013)

[158] Shameli-Sendi, A., Aghababaei-Barzegar, R., Cheriet, M.: Taxonomy of information security risk assessment (isra). Computers & security **57**, 14–30 (2016)

[159] Shevchenko, N., Chick, T.A., O'Riordan, P., Scanlon, T.P., Woody, C.: Threat modeling: a summary of available methods. Tech. rep., Carnegie Mellon University Software Engineering Institute Pittsburgh United ... (2018)

[160] Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing, J.M.: Automated generation and analysis of attack graphs. In: Proceedings 2002 IEEE Symposium on Security and Privacy. pp. 273–284. IEEE (2002)

[161] Shi, Z., Graffi, K., Starobinski, D., Matyunin, N.: Threat modeling tools: A taxonomy. IEEE Security & Privacy **20**(4), 29–39 (2021)

[162] Shostack, A.: Threat modeling: Designing for security. John Wiley & Sons (2014)

[163] Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requirements engineering **10**, 34–44 (2005)

[164] Slovic, P.: The construction of preference. American psychologist **50**(5), 364 (1995)

[165] Stevens, R., Votipka, D., Redmiles, E.M., Ahern, C., Sweeney, P., Mazurek, M.L.: The battle for new york: A case study of applied digital threat modeling at the enterprise level. In: 27th USENIX Security Symposium (USENIX Security 18). pp. 621–637 (2018)

[166] Stølen, K., Erdogan, G.: The coras tool, https://coras.sourceforge.net/coras_tool.html

[167] Talhi, C., Mouheb, D., Lima, V., Debbabi, M., Wang, L., Pourzandi, M.: Usability of security specification approaches for uml design: A survey. J. Object Technol. **8**(6), 102–122 (2009)

[168] Tatam, M., Shanmugam, B., Azam, S., Kannoorpatti, K.: A review of threat modelling approaches for apt-style attacks. Heliyon **7**(1) (2021)

[169] The Organization for Economic Cooperation and Development: Enterprises by business size - oecd data. https://doi.org/10.1787/31d5eeaf-en, https://data.oecd.org/entrepreneur/enterprises-by-business-size.htm

[170] Threatmodeler: Threatmodeler® software, https://threatmodeler.com/threatmodeler/#threatmodeler

[171] Tøndel, I.A., Jensen, J., Røstad, L.: Combining misuse cases with attack trees and security activity models. In: 2010 International Conference on Availability, Reliability and Security. pp. 438–445. IEEE (2010)

[172] Torr, P.: Demystifying the threat modeling process. IEEE Security & Privacy **3**(5), 66–70 (2005)

[173] Triantaphyllou, E., Triantaphyllou, E.: Multi-criteria decision making methods. Springer (2000)

[174] Tudor, D., Walter, G.A.: Using an agile approach in a large, traditional organization. In: AGILE 2006 (AGILE'06). pp. 7–pp. IEEE (2006)

[175] Tuma, K., Calikli, G., Scandariato, R.: Threat analysis of software systems: A systematic literature review. Journal of Systems and Software **144**, 275–294 (2018)

[176] Tzeng, G.H., Lin, C.W., Opricovic, S.: Multi-criteria analysis of alternative-fuel buses for public transportation. Energy policy **33**(11), 1373–1383 (2005)

[177] UcedaVelez, T., Morana, M.M.: Risk Centric Threat Modeling: process for attack simulation and threat analysis. John Wiley & Sons (2015)

[178] Uma, M., Padmavathi, G.: A survey on various cyber attacks and their classification. Int. J. Netw. Secur. **15**(5), 390–396 (2013)

[179] Uzunov, A., Fernandez, E., Falkner, K.: Engineering security into distributed systems: A survey of methodologies. Journal of Universal Computer Science **18**(20), 2920–3006 (2012)

[180] Uzunov, A.V., Fernandez, E.B.: An extensible pattern-based library and taxonomy of security threats for distributed systems. Computer Standards & Interfaces **36**(4), 734–747 (2014)

[181] Van Horenbeek, A., Pintelon, L.: Development of a maintenance performance measurement framework—using the analytic network process (anp) for maintenance performance indicator selection. Omega **42**(1), 33–46 (2014)

[182] Van Lamsweerde, A.: Elaborating security requirements by construction of intentional anti-models. In: Proceedings. 26th International Conference on Software Engineering. pp. 148–157. IEEE (2004)

[183] Van Lamsweerde, A., Letier, E.: Handling obstacles in goal-oriented requirements engineering. IEEE Transactions on software engineering **26**(10), 978–1005 (2000)

[184] Van Lamsweerde, A., et al.: Engineering requirements for system reliability and security. NATO Security Through Science Series D-Information and Communication Security **9**, 196 (2007)

[185] Van Landuyt, D., Joosen, W.: A descriptive study of assumptions in stride security threat modeling. Software and Systems Modeling pp. 1–18 (2021)

[186] Vidalis, S., Jones, A.: Analyzing threat agents and their attributes. In: ECIW. pp. 369–380. Citeseer (2005)

[187] Villarroel, R., Fernández-Medina, E., Piattini, M.: Secure information systems development–a survey and comparison. Computers & Security **24**(4), 308–321 (2005)

[188] Wang, Y., Ruhe, G.: The cognitive process of decision making. International Journal of Cognitive Informatics and Natural Intelligence (IJCINI) **1**(2), 73–85 (2007)

[189] Wangen, G.: Information security risk assessment: a method comparison. Computer **50**(4), 52–61 (2017)

[190] Wangen, G., Hallstensen, C., Snekkenes, E.: A framework for estimating information security risk assessment method completeness: Core unified risk framework, curf. International Journal of Information Security **17**, 681–699 (2018)

[191] Ware, M.S., Bowles, J.B., Eastman, C.M.: Using the common criteria to elicit security requirements with use cases. In: Proceedings of the IEEE SoutheastCon 2006. pp. 273–278. IEEE (2006)

[192] Wątróbski, J., Jankowski, J., Ziemba, P., Karczmarczyk, A., Zioło, M.: Generalised framework for multi-criteria method selection. Omega **86**, 107–124 (2019)

[193] van de Weerd, I., Brinkkemper, S.: Meta-modeling for situational analysis and design methods. In: Handbook of research on modern systems analysis and design technologies and applications, pp. 35–54. IGI Global (2009)

[194] Whittle, J., Wijesekera, D., Hartong, M.: Executable misuse cases for modeling security concerns. In: Proceedings of the 30th international conference on Software engineering. pp. 121–130 (2008)

[195] Wieringa, R.J.: Design science methodology for information systems and software engineering. Springer (2014)

[196] Williams, I., Yuan, X.: Evaluating the effectiveness of microsoft threat modeling tool. In: Proceedings of the 2015 information security curriculum development conference. pp. 1–6 (2015)

[197] Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th international conference on evaluation and assessment in software engineering. pp. 1–10 (2014)

[198] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in software engineering. Springer Science & Business Media (2012)

[199] Wright, M., Chizari, H., Viana, T.: A systematic review of smart city infrastructure threat modelling methodologies: a bayesian focused review. Sustainability **14**(16), 10368 (2022)

[200] Xiong, W., Lagerström, R.: Threat modeling–a systematic literature review. Computers & security **84**, 53–69 (2019)

[201] Xu, D., Nygard, K.E.: Threat-driven modeling and verification of secure software using aspect-oriented petri nets. IEEE transactions on software engineering **32**(4), 265–278 (2006)

[202] Yeh, W.C., Chuang, M.C.: Using multi-objective genetic algorithm for partner selection in green supply chain problems. Expert Systems with applications **38**(4), 4244–4253 (2011)

[203] Yskout, K., Heyman, T., Van Landuyt, D., Sion, L., Wuyts, K., Joosen, W.: Threat modeling: from infancy to maturity. In: 2020 IEEE/ACM 42nd International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER). pp. 9–12. IEEE (2020)

[204] Zografopoulos, I., Ospina, J., Liu, X., Konstantinou, C.: Cyber-physical energy systems security: Threat modeling, risk assessment, resources, metrics, and case studies. IEEE Access **9**, 29775–29818 (2021)

A Validation check

Table A.1: Record of the proposed validity check.

Term	Related experimental information	Concern
	Conclusion validity	
Low statistical power	Data gathering is conducted by doing an SLR, surveys, and interviews. Although the sample size of the survey has low statistical power, this is mitigated by the strict selection of participants and doing follow-up interviews to supplement the results with qualitative data. The amount of interviews conducted is expected to yield more than an 80% saturation [46].	No
Violated assumptions of statistical tests	Only the average is used in order to draw conclusions. So no statistical assumptions are violated.	No
Fishing and the error rate	Manual data processing is used in the Systematic Literature Review, which enables fishing. The research is executed in combination with an internship at the case study company. This bias is mitigated through the design of the study. By first establishing a literature base and refining this with experts, it is less likely that fishing is performed. In addition, the researcher does not gain anything from reporting a specific result	No
Reliability of measures	Following the framework we gathered and organized knowledge from various sources of knowledge. All refinements steps have been reported on. The requirements that were used in sever steps and were improved throughout the research are still multi-interpretable according to the case study.	Yes
Reliability of treatment implementation	The treatment is adapted from a published and peer reviewed framework. The treatment is equally applied for all participants. However, these participants are part of a singlegroup decision-making process. To make the treatment more reliable, multiple future case studies should be performed. Additionally, not all participants performed the preparation activity equally, which can harm the reliability.	Yes
Random irrelevancies in experimental settings	The interviews and case study were performed in a closed (online) environment, it is not expected to have any major external disruptions.	No
Random heterogeneity of subjects	The subjects for the expert interview were scattered over multiple organisations and countries which increases heterogeneity. They are selected through purposive sampling and snowballing, which may harm the heterogeneity. The participants of the case studies were a group of decision-makers from the same organisation. This causes an increase in homogeneity, which reduces the external validity, but does not harm conclusion validity.	Yes

Internal validity		
History	This study is performed a few years after a worldwide pandemic. However, this is not expected to have any influence on this study.	No
Maturation	There is no direct maturation effect within the case study because of the relatively short time span. The participants of the surveys and interviews are selected based on being familiar with threat modeling. Although the participants' work experience has been verified, the study did not differentiate between different levels of experience with threat modeling method selection.	Yes
Testing	Testing the approach is only done once, there is no repetition. So there is no possible testing bias.	No
Instrumentation	The instrumentation used in the survey was suggested to be exhaustive. Requirements were found to be multi-interpretable. To deal with this, requirements have been refined after each research activity. Nevertheless, some improvements are still indicated to be done.	Yes
Statistical regression	Subjects are not classified into groups based on a previous experiment. Survey candidates were not subjected to an interview only if they denied the opportunity.	No
Selection	The selection of expert participants is done through purposive sampling[19] and by means of snowballing. The selection of case study participants is executed by means of convenience. Due to the lack of resources and connections, this bias can not be mitigated.	Yes
Mortality	Participants can withdraw their consent anytime. In that case, the data gathered from the person will be disregarded. This did not occur.	No
Ambiguity about the direction of causal influence	The causal influence used in this study is validated by Farshidi [51].	No
Interactions within selection	Interactions within the selection are not expected to create a bias. Since the participants of the case study are from the same teams they work with on a daily basis. Also, this study does not contain a multiple-group experiment. Still, this bias should be monitored when future case studies are conducted.	No
Diffusion or imitation of treatments	Only a single case study is performed. There is potential for opinion affecting interaction between participants, given that they are from the same organisation and perform the preparation asynchronously.	No
Compensatory equalization of treatments	Participants don't get monetary compensation for participating in the interview or case study.	No
Compensatory rivalry	This threat does not apply, since there are no monetary compensations.	No
Resentful demoralization	Al participants get the same treatment, so there is no possibility of resentful demoralization.	No
Construct validity		
Inadequate explication of constructs	Research proposal is to be approved by the supervisors. Each construct has been reviewed by the supervisors before deploying. A pilot of the interview and survey was done in order to identify constructs that need additional clarification.	No
Mono-operation bias	Only a single case study is performed. This case may not be representative of the population. Due to resource constraints, performing multiple case studies is future work. This is iteratively mentioned throughout the report.	Yes

Mono-method bias	A variety of data gathering methods are used including SLR, survey, and expert interviews. Furthermore the artifact created based on these data gathering methods is tested in a case study. A decision model is an abstract of reality, so it is not expected to capture the complete threat modeling method selection spectrum.	No
Confounding constructs and levels of constructs	The researcher is not aware of any missed confounding constructs that affect the decision making process. The applied framework incorporates preference by using a qualitative prioritization of decision-making criteria.	No
Interaction of different treatments	There is only a single treatment applied.	No
Interaction of testing and treatment	The case study does a post-task questionnaire to test a carefully selected set of multiple evaluation criteria. Because this test is done afterward, it does not have any interaction with the treatment. However, it can be foreseen that participating in the survey has an influence on how the treatment is perceived.	Yes
Restricted generalizability across constructs	A set of constructs is selected which the case study participants evaluate on. Although, this set is not complete it covers some potential negative effects.	No
Evaluation apprehension	Evaluation apprehension is mitigated by providing information regarding the goals of the study and the method for processing the data (anonymous). Furthermore, it is made clear that individual performance does not play a part in the evaluation. The focus of the study is upon the threat modeling methods and the systematic decision making approach.	No
Experimenter expectancies	There is room for ambiguity throughout all the interview components of the experiment. This is mitigated by checking the prepared questions for such bias. Although, the case study is guided by the researcher, the outcome expectancy is not expected to have influenced the prioritization of the participants.	Yes
Hypothesis guessing	The participants of the case study expected that the outcome of the case study was a one tool solution. In addition, it was expected that the decision-model also determined if threat modeling was relevant in the context of the organisation. However, the outcome was a ranking of multiple threat modeling methods, constructed of a technique, notation, and a tool. The relevancy was only determined in terms of requirement alignment and not based on what is already implemented. This may have negatively influenced the perception of the outcome, because it did not align with the expected outcomes.	Yes
	External validity	
Interaction of selection and treatment	Due to the limited availability of participants that are familiar with threat modeling, participants from the expert surveys and interviews were working in a variety of organisations. The case study is only done in one large organisation, which harms the generalizability.	Yes
Interaction of setting and treatment	The experiment is performed in a business setting, where threat modeling is already introduced. Here, there are dedicated security information officers that can make a group decision. When companies are smaller, there might be a single representative with a set of combined roles.	No
Interaction of history and treatment	The threat modeling domain is continuously developing. The decision model is created for a current snapshot of time. When repeating this study in the future an updated knowledge base needs to be established.	No

B Survey results

B.1 Quality ranking

Table B.1: The relative ranking of the quality groups by domain experts. Rank # indicates the position of the quality compared to the other items.

	Rank #1		**Rank #2**		**Rank #3**	
Quality criteria	Percentage	Count	Percentage	Count	Percentage	Count
Efficiency	35.71%	5	21.43%	3	42.86%	6
Reliability	21.43%	3	35.71%	5	42.86%	6
Applicability	42.86%	6	42.86%	6	14.29%	2

Table B.2: The relative ranking of the quality criteria within the efficiency group. Rank # indicates the position of the quality compared to the other items.

	Rank #1		**Rank #2**		**Rank #3**		**Rank #4**	
Quality criteria	Percentage	Count	Percentage	Count	Percentage	Count	Percentage	Count
Cost	35.71%	5	21.43%	3	14.29%	2	28.57%	4
Reusability	21.43%	3	21.43%	3	42.86%	6	14.29%	2
Tailorability	21.43%	3	21.43%	3	14.29%	2	42.86%	6
Scalability	21.43%	3	35.71%	5	28.57%	4	14.29%	2

Table B.3: The relative ranking of the quality criteria within the Reliability group. Rank # indicates the position of the quality compared to the other items.

	Rank #1		**Rank #2**		**Rank #3**		**Rank #4**		**Rank #5**		**Rank #6**		**Rank #7**		**Rank #8**	
Quality criteria	Percentage	Count	Percentage	Count	Percentage	Count	Percentage	Count	Percentage	Count	Percentage	Count	Percentage	Count	Percentage	Count
Completeness	7.14%	1	7.14%	1	7.14%	1	7.14%	1	14.29%	2	21.43%	3	21.43%	3	14.29%	2
Precision (Non-ambiguity)	0.00%	0	7.14%	1	21.43%	3	14.29%	2	21.43%	3	14.29%	2	7.14%	1	14.29%	2
Documentation	0.00%	0	7.14%	1	0.00%	0	14.29%	2	14.29%	2	14.29%	2	21.43%	3	28.57%	4
Software evolution support	7.14%	1	21.43%	3	0.00%	0	35.71%	5	7.14%	1	21.43%	3	0.00%	0	7.14%	1
Suitability	28.57%	4	7.14%	1	35.71%	5	7.14%	1	14.29%	2	0.00%	0	0.00%	0	7.14%	1
Maturity	7.14%	1	14.29%	2	0.00%	0	0.00%	0	0.00%	0	14.29%	2	42.86%	6	21.43%	3
Support for maintainability	0.00%	0	14.29%	2	14.29%	2	14.29%	2	28.57%	4	14.29%	2	7.14%	1	7.14%	1
Understandability	50.00%	7	21.43%	3	21.43%	3	7.14%	1	0.00%	0	0.00%	0	0.00%	0	0.00%	0

Table B.4: The relative ranking of the sub-quality criteria within the reliability group, as a sub-quality of Documentation. Rank # indicates the position of the quality compared to the other items.

	Rank #1		**Rank #2**	
Quality criteria	Percentage	Count	Percentage	Count
Technique documentation	85.71%	12	14.29%	2
Tool documentation	14.29%	2	85.71%	12

Table B.5: The relative ranking of the sub-quality criteria within the reliability group, as a sub-quality of Software evolution support. Rank # indicates the position of the quality compared to the other items.

	Rank #1		**Rank #2**		**Rank #3**		**Rank #4**	
Quality criteria	Percentage	Count	Percentage	Count	Percentage	Count	Percentage	Count
Modularity	21.43%	3	7.14%	1	35.71%	5	35.71%	5
Component architecture	50.00%	7	21.43%	3	21.43%	3	7.14%	1
Change propagation	21.43%	3	14.29%	2	28.57%	4	35.71%	5
Change impact analysis	7.14%	1	57.14%	8	14.29%	2	21.43%	3

Table B.6: The relative ranking of the sub-quality criteria within the reliability group, as a sub-quality of Maturity. Rank # indicates the position of the quality compared to the other items.

	Rank #1		**Rank #2**	
Quality criteria	Percentage	Count	Percentage	Count
Technique maturity	78.57%	11	21.43%	3
Tool maturity	21.43%	3	78.57%	11

Table B.7: The relative ranking of the sub-quality criteria within the reliability group, as a sub-quality of Support for maintainability. Rank # indicates the position of the quality compared to the other items.

	Rank #1		**Rank #2**	
Quality criteria	Percentage	Count	Percentage	Count
Modifiable	71.43%	10	28.57%	4
Traceability	28.57%	4	71.43%	10

Table B.8: The relative ranking of the quality criteria within the applicability group. Rank # indicates the position of the quality compared to the other items.

	Rank #1		**Rank #2**		**Rank #3**	
Quality criteria	Percentage	Count	Percentage	Count	Percentage	Count
User experience	42.86%	6	50.00%	7	7.14%	1
Compatibility with agile development process	42.86%	6	28.57%	4	28.57%	4
Compatbility with related processes	14.29%	2	21.43%	3	64.29%	9

Table B.9: The relative ranking of the sub-quality criteria within the applicability group, as a sub-quality of user experience. Rank # indicates the position of the quality compared to the other items.

	Rank #1		**Rank #2**	
Quality criteria	Percentage	Count	Percentage	Count
Ease of use	50.00%	7	50.00%	7
Learning curve	50.00%	7	50.00%	7

B.2 Relevancy and applicability of requirements

Table B.10: The relevancy of requirements and applicability of user stories by domain experts. Original denotes the original requirement. R stands for Relevant, A stands for Applicable, NR denotes Non-Relevant and NO means No-Opinion. The percentage indicates that the column contains the cumulative percentage.

ID	Original	User story - Description	R/A	R/A%	NR	NR%	NO	NO%
E001	Input		*No evaluation needed*					
US101	Current security [157, 158]	As a decision-maker, I want the method to take current security measures into account, so irrelevant threats can be disregarded. (E.g. DDOS threat disregarded due to DDOS protection implemented)	9	64.29%	5	35.71%	0	0.00%
US102	Input type [157, 158, 175]	What type of input required in order to start the threat analysis.	13	92.86%	1	7.14%	0	0.00%
a		As a decision-maker, I want the analysis to require an attacker behavior[A] specification, so the analysis is based on what the system should be protected from.	11	78.57%				
b		As a decision-maker, I want the analysis to require security assumptions[B], so the analysis is based on what the system should be protected from.	9	64.29%				
c		As a decision-maker, I want the analysis to require system goals[C], so that the analysis can be performed as soon as a high-level description is provided.	10	71.43%				
d		As a decision-maker, I want the analysis to require the architectural design of the system, so that the analysis is performed based on a model of the system.	12	85.71%				
e		As a decision-maker, I want the analysis to be done based-on source code, so that the analysis is performed based on the actual system.	4	28.57%				
US103	Input data [29]	Where the input data comes from.	10	71.43%	4	28.57%	0	0.00%
a		As a decision-maker, I want the threat assessment be based-on expert opinion, so that no additional input data is required.	8	57.14%				
b		As a decision-maker, I want the threat assessment to be done based-on historical data.	6	42.86%				
E002	Output		*No evaluation needed*					
US201	Output type [107, 140, 158]	The type of artifact that the method produces.	13	92.86%	1	7.14%	0	0.00%
a		As a decision-maker, I want the threat modeling method to produce security threats.	13	92.86%				

b		As a decision-maker, I want the threat modeling method to produce threat mitigations, so general risk-lowering countermeasures are obtained.	12	85.71%				
c		As a decision-maker, I want the threat modeling method to produce security requirements, so that a refined set of specific countermeasures is obtained.	10	71.43%				
US202	Output granularity [175]	The level of detail in which the result from threat modeling is presented.	13	92.86%	1	7.14%	0	0.00%
a		As a decision-maker, I want the threat modeling method to produce a high-level outcome, so a general overview can be obtained that can guide the action points. (An example of a high level outcome is a document that provides an overview of general security improvements.)	13	92.86%				
b		As a decision-maker, I want the threat modeling method to produce a low-level outcome, so that the outcome can be directly used for implementation. (An example of a low-level outcom is a model of the improved system, which can directly be translated into code.)	10	71.43%				
US203	Report representation [22, 175]	How the findings from threat modeling are displayed in the report.	12	85.71%	2	14.29%	0	0.00%
a		As a decision-maker, I want the report from a threat modeling project to-be a structured text.	6	42.86%				
b		As a decision-maker, I want the report from a threat modeling project to have a model-based representation.	10	71.43%				
US204	Mitigation strategy [159]	As a decision-maker, I want the method to assist in providing mitigation strategies.	11	78.57%	3	21.43%	0	0.00%
E003	Perspective		*No evaluation needed*					
US301	Approach [57, 107, 168, 175]	The central ideology that the threat modeling method is based on.	13	92.86%	1	7.14%	0	0.00%
a		As a decision-maker, I want the technique to be attack-centric, so there will be a focus on identifying attacker profiles and the complexity of attacks.	6	42.86%				
b		As a decision-maker, I want the technique to be risk-centric, so that impact and likelihood of the threats decide which security requirement needs to be addressed first.	9	64.29%				
c		As a decision-maker,I want the technique to be software-centric, so the focus will be on the software that is examined. (E.g. STRIDE analysis on DFDs)	8	57.14%				
d		As a decision-maker, I want the technique to be centered around Goal-Oriented Requirement Engineering(GORE), so that both functional and non-functional (among other, security) requirements are obtained.	5	35.71%				

e		As a decision-maker, I want the technique to be centered around Security Requirements Engineering (SRE), so that the primary goal is to identify security requirements.	4	28.57%				
US302	Focus [159]	The viewpoint from which the technique is performed.	10	71.43%	4	28.57%	0	0.00%
a		As a decision-maker, I want the technique to be designed from the point of view of the attacker.	10	71.43%				
b		As a decision-maker, I want the technique to be designed from the point of view of a defender.	11	78.57%				
US303	Agents [49]	Does a threat model focus on one perspective or multiple perspectives at the same time.	12	85.71%	1	7.14%	1	7.14%
a		As a decision-maker, I want one model to show the perspectives of multiple agents so that conflicting perspectives can be considered. (For attack-centric approaches)	11	78.57%				
b		As a decision-maker, I want one model to show the perspective of a single agent, so that the model does not become too complicated. (For attack-centric approaches)	3	21.43%				
US304	Modeling view [43]	The level of detail on which the actual model is created.	14	100%	0	0.00%	0	0.00%
a		As a decision-maker, I want the threat model to describe the architecture of the system, so that threats can be related to system components.	13	92.86%				
b		As a decision-maker, I want the threat model to describe the functional behavior[D] of the system. (E.g. Use state diagrams to do threat modeling)	11	78.57%				
c		As a decision-maker, I want the threat model to describe the platform of the system, so that operating systems, middleware and also physical parts are considered.	8	57.14%				
E004	Modeling notation					*No evaluation needed*		
US401	Notation Abstraction [86, 158]	The level of detail at which the components are represented in the notation.	13	92.86%	1	7.14%	0	0.00%
a		As a decision-maker, I want the model notation to represent its components centered around (security) goals[E].	5	35.71%				
b		As a decision-maker, I want the model notation to represent its components centered around a model of the system.	8	57.14%				
c		As a decision-maker, I want the model notation to represent its components centered around (security) problems[F].	5	35.71%				
d		As a decision-maker, I want the model notation to represent its components centered around the business processes.	4	28.57%				
US402	Formality [175, 200]	Distinguishes between formal (textual) and graphical representations.	12	85.71%	1	7.14%	1	7.14%
a		As a decision-maker, I want the notation to be formal, so that threat modeling based on a mathematical model is possible.	1	7.14%				
b		As a decision-maker, I want the notation to be graphical, so that attack trees, attack graphs, Data Flow Diagrams (DFDs), or tables can be used.	14	100%				

US403	Language [22, 50, 57, 107]	As a decision-maker, I want to use a notation for threat modeling that I am familiar with. (E.g. DFD, petri-nets, UML)	12	85.71%	1	7.14%	1	7.14%
US404	Similarity with software specification languages [34]	As a decision-maker, I want to use (an extension of) a notation that is a well-known software specification language, so that threat modeling becomes easier to learn.	9	64.29%	3	21.43%	2	14.29%
US405	Extensibility	As a decision-maker, I want to be able to extend the modeling notation, so that domain-specific components can be modeled.	10	71.43%	4	28.57%	0	0.00%
US406	Coverage of components [34]	As a decision-maker, I want the notation to have a specific set of components present.	10	71.43%	4	28.57%	0	0.00%
US407	Relationships between components [34]	As a decision-maker, I want to have a notation where the most common relationships between components are present.	9	64.29%	4	28.57%	1	7.14%
US408	Countermeasure impact [45]	As a decision-maker, I want to be able to visualize how the proposed countermeasures affect the system's weaknesses, so I can understand how effective the mitigations are.	9	64.29%	4	28.57%	1	7.14%
E005	Modeling context		*No evaluation needed*					
US501	Barriers for practitioners [157]	As a decision-maker, I want a threat modeling method that does not have any knowledge barriers[G].	10	71.43%	3	21.43%	1	7.14%
US502	Layer [140]	The level of system details at which threats are assessed.	14	100%	0	0.00%	0	0.00%
a		As a decision-maker, I want the threat modeling method to help identify threats on the software level, so that only software components are considered.	8	57.14%				
b		As a decision-maker, I want the threat modeling method to help identify threats on the computer-based system level, so that both software and hardware are considered.	9	64.29%				
c		As a decision-maker, I want the threat modeling method to help identify threats on the total system level, so that man, technology and organisation (MTO) are considered.	12	85.71%				
d		As a decision-maker, I want the threat modeling method to help identify threats on the environment level, so that factors beyond MTO are considered.	9	64.29%				
US503	Applicability to system [50, 140, 159, 200]	The method is designed for either general use or a specific type of computer-based system.	11	78.57%	1	7.14%	2	14.29%

a		As a decision-maker, I want to have a threat modeling method that is generally applicable, so I can use it in many different systems.	12	85.71%				
b		As a decision-maker,I want to have a method that is specifically designed for the system under analysis. (An example of a specific method would be: a threat modeling method for cloud applications)	7	50.00%				
US504	Hardware threats [159]	As a decision-maker, I want to have a threat modeling method that considers threats to hardware. (E.g. physical tempering is considered)	8	57.14%	4	28.57%	2	14.29%
E006	Method process		*No evaluation needed*					
US601	Threat generation [22, 168, 200]	The process of identifying and establishing a list of potential threats.	12	85.71%	2	14.29%	0	0.00%
a		As a decision-maker, I want to have a method that generates threats based on a knowledge base/methodology. (e.g. Threat identification using STRIDE)	12	85.71%				
b		As a decision-maker, I want to have a threat modeling method that generates threats based on expert knowledge. (e.g. CORAS)	5	35.71%				
US602	Involved entities [157]	As a decision-maker, I want to have a threat modeling method that supports the participation of specific stakeholders. (e.g. Threat modeling by security specialist versus threat modeling by developer)	13	92.86%	1	7.14%	0	0.00%
US603	Set boundary [157]	As a decision-maker, I want the technique to clearly explain how to define the limits of the system, so that we can identify the area that needs protection. (E.g. only consider assets with a specific priority level)	7	50.00%	5	35.71%	2	14.29%
E007	Analysis		*No evaluation needed*					
US701	Resource valuation [157, 158]	As a decision-maker, I want a threat modeling method that differentiates between critical and non-critical components.	10	71,43%	4	28,57%	0	0,00%
US702	Effect propagation [45, 158]	I want the threat modeling method to show how attacks on one part of the system can affect other parts that depend on it. (e.g. internet facing login application is compromised –>effect on user database)	12	85,71%	2	14,29%	0	0,00%
US703	Quality assurance [49, 130, 175]	I want a technique that ensures the quality of the outcome as part of the analysis procedure. (e.g. Use guidelines for assessing the quality of the model)	9	64,29%	4	28,57%	1	7,14%
US704	Threat library [28, 175, 201]	As a decision-maker, I want a method that uses a list of known threats (a threat library) to identify applicable security threats based on the system's components.	11	78,57%	3	21,43%	0	0,00%
US705	Analysis level [3]	The level of detail used to analyze the system for potential threats.	13	92,86%	1	7,14%	0	0,00%

a		As a decision-maker, I want the depth of the analysis to reach source code level.	3	21,43%				
b		As a decision-maker, I want the depth of the analysis to reach system design level, so it can produce a detailed outcome. (System design may include, technical details of security controls such as encryption standards, etc.)	12	85,71%				
c		As a decision-maker, I want the depth of the analysis to reach architecture level, so it can be executed early on in the development process.	13	92,86%				
US706	Security objective [49, 50, 107, 130, 175]	The areas of security covered by the method.	14	100%	0	0,00%	0	0,00%
a		As a decision-maker, I want the security objective of the threat modeling method to cover confidentiality. (In other words, the threat modeling method can analyse threats to confidentiality)	13	92,86%				
b		As a decision-maker, I want the security objective of the threat modeling method to cover integrity.	13	92,86%				
c		As a decision-maker, I want the security objective of the threat modeling method to cover availability.	13	92,86%				
US707	Prioritization [45]	As a decision-maker, I want the technique to prioritize threats, so that we can determine the order in which to implement the countermeasures.	10	71,43%	3	21,43%	1	7,14%
US708	Risk [175]	The role of risk in the threat modeling technique. Risk is expressed in terms of likelihood and impact of a malicious event.	12	85,71%	1	7,14%	1	7,14%
a		As a decision-maker, I want to have a threat modeling technique that associates risk (Likelihood & impact) to the identified threats, so that threats can be prioritized	11	78,57%				
b		As a decision-maker, I want to have a threat modeling technique that considers risk (Likelihood & impact) externally. (E.g. by combining the technique with an external risk management framework)	7	50,00%				
c		As a decision-maker, I want to have a threat modeling technique that does not consider risk. (E.g. because it is not needed to obtain the prefered threat modeling outcome)	3	21,43%				
US709	Steps of vulnerability exploitation [45]	As a decision-maker, I want to have a threat modeling method that considers steps to vulnerability exploitation[H] (expressed in time) as part of the analysis.	8	57,14%	5	35,71%	1	7,14%
US710	Threat analysis type [28, 29, 200]	The differentiation between quantitative and qualitative techniques for threat analysis.	8	57,14%	4	28,57%	2	14,29%
a		As a decision-maker, I want my threat modeling analysis to be qualitative[I], so it is faster.	9	64,29%				

b		As a decision-maker, I want my threat modeling method to be quantitative[J], so it is more detailed.	1	7,14%				
E008	Other		*No evaluation needed*					
US801	Stopping condition [175]	As a decision-maker, I want to have a definition of done (stopping condition) in the technique, so users are uniformly guided on when the threat modeling is done.	10	71.43%	3	21.43%	1	7.14%
US802	Interoperability [140]	As a decision-maker, I want a threat modeling method that can work together with another cyber security method. (E.g. Can the results of threat modeling be used in another security method?, or is the threat modeling method a part of a group of security methods?)	10	71.43%	3	21.43%	1	7.14%
US803	Validation [34, 117]	As a decision-maker, I want to receive validation support for the threat model, so that I can be more sure that it accurately reflects the real-world situation. (e.g. Tool provides recommendations and feedback about common mistakes during the model development).	11	78.57%	3	21.43%	0	0.00%
US804	Verification [34, 117]	As a decision-maker, I want to receive verification support for the threat model, so I can guarantee the model is built correctly. (E.g. Only allow relationships between components when they are meaningful)	8	57.14%	3	21.43%	3	21.43%
US805	Automation [159]	As a decision-maker, I want to have a tool that assists the threat modeling technique by (semi-)automating some of the steps.	11	78.57%	1	7.14%	2	14.29%
US806	Portability [159]	As a decision-maker, I want a tool that can transfer projects from one device to another. (For example, through an export and import function.)	13	92.86%	1	7.14%	0	0.00%

C Interview analyses

C.1 Requirement occurrence analysis

Table C.1: The considered requirements by the interviewees. Checkmarks denote the requirements mentioned by the domain experts, and cross marks signify that the domain expert did not mention the requirement. The Mention rate denotes the unique mentions of the requirement over all the interviews. The total mentions indicate the number of segments in which the requirement has been detected.

ID	**Requirement**	**Mention rate**	**Total mentions**	Interview 1	Interview 2	Interview 3	Interview 4	Interview 5	Interview 6	Interview 7	Interview 8	Interview 9	Interview 10	Interview 11	Interview 12
E001: Input															
US101	Current security	50,00%	9	✓	✗	✗	✓	✓	✗	✗	✓	✓	✗	✓	✗
US102	Input type	75,00%	30	✓	✓	✗	✓	✓	✓	✗	✓	✓	✗	✓	✓
US102a	Attacker behavior	33,33%	5	✗	✓	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗
US102b	Security assumptions	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US102c	System goals	16,67%	2	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
US102d	Architectural design	58,33%	16	✓	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
US102e	Source code	25,00%	3	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗
US102f	Business processes (New)	16,67%	2	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
US102g	Descriptive language (New)	16,67%	4	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
US103	Input data	75,00%	26	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗
US103a	Expert opinion	41,67%	6	✗	✗	✓	✗	✗	✓	✗	✓	✓	✗	✓	✗
US103b	Historical data	50,00%	9	✗	✓	✓	✗	✗	✓	✓	✓	✗	✗	✓	✗
US103c	Threat intelligence tooling (New)	16,67%	3	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
US103d	Detailed threats (New)	16,67%	2	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
US103e	MITRE (New)	8,33%	1	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
US103f	Personas (New)	25,00%	4	✗	✗	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗
US103g	Security cards (New)	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
E002: Output															
US201	Output type	66,67%	16	✗	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓	✓

US201a	Security threats	16,67%	3	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗
US201b	Threat mitigations	25,00%	5	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✓	✗
US201c	Security requirements	8,33%	1	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
US201d	Security tickets	41,67%	7	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓
US202	Output granularity	66,67%	17	✗	✗	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓
US202a	High-level outcome	50,00%	11	✗	✗	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗
US202b	Low-level outcome	41,67%	8	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓	✓
US203	Report representation	50,00%	8	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	✓
US203a	Structured text	8,33%	1	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
US203b	Model-based	50,00%	7	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	✓
US204	Mitigation strategy	16,67%	2	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
US205	Security patterns (New)	41,67%	8	✓	✗	✓	✓	✗	✗	✗	✓	✗	✓	✗	✗
US206	Actionable findings (New)	66,67%	16	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗	✗
US207	Common vulnerabilities (New)	25,00%	3	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗	✗
E003:	**Perspective**														
US301	Approach	66,67%	15	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗
US301a	Attack-centric	8,33%	1	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
US301b	Risk-centric	50,00%	9	✓	✗	✗	✓	✓	✗	✓	✓	✗	✗	✓	✗
US301c	Software-centric	25,00%	7	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗
US301d	GORE	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US301e	SRE	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US302	Focus	33,33%	4	✗	✗	✗	✗	✓	✗	✓	✓	✗	✓	✗	✗
US302a	Attacker view	33,33%	4	✗	✗	✗	✗	✓	✗	✓	✓	✗	✓	✗	✗
US302b	Defender view	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US303	Agents	25,00%	3	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗
US303a	Multiple agents	25,00%	3	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗
US303b	Single agent	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US304	Modeling view	33,33%	7	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✓	✗
US304a	System architecture	33,33%	6	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✓	✗
US304b	Functional behavior	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US304c	Platform of the system	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
US304d	Process (New)	8,33%	2	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
E004:	**Modeling notation**														
US401	Notation abstraction	16,67%	2	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
US401a	Goal-centric	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US401b	Model centric	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
US401c	Problem-centric	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US401d	Process-centric	8,33%	1	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
US402	Formality	33,33%	9	✓	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗
US402a	Mathematical model	8,33%	1	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
US402b	Graphical	33,33%	8	✓	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗
US403	Language	50,00%	9	✗	✓	✗	✗	✓	✓	✗	✓	✓	✓	✗	✗
US404	Similarity with software specification languages	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
US405	Notation Extensibility	16,67%	3	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
US406	Coverage of components	16,67%	2	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
US407	Relationshipls between components	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
US408	Countermeasure impact	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗

US409	Priority visualization (New)	16,67%	2	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
E005: Modeling context															
US501	Barriers for pracitioners	25,00%	3	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
US502	Layer - Level of sytem details	25,00%	3	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓
US502a	Software level	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
US502b	Computer-based system level	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US502c	Total system level	16,67%	2	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
US502d	Environment level	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US503	Applicability to system	33,33%	4	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗
US503a	General appicability	33,33%	4	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗
US503b	Specific applicability	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US504	Hardware threats	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US505	Layer decomposition	33,33%	5	✗	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗
E006: Method process															
US601	Threat generation	75,00%	29	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗
US601a	Knowledge / methodology based	50,00%	17	✗	✓	✓	✓	✗	✓	✗	✗	✓	✓	✗	✗
US601b	Expert knowlege based	50,00%	9	✗	✗	✓	✗	✓	✓	✗	✗	✓	✓	✓	✗
US601c	User knowledge based (New)	66,67%	19	✗	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✗
US602	Involved entities	33,33%	6	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓
US603	Set boundary to identify area of protection	25,00%	3	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓	✗
US604	Unique Threats (New)	8,33%	3	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US605	Trigger (New)	33,33%	9	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗
US605a	Static time-slots (New)	16,67%	3	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
US605b	Dynamic sessions (New)	33,33%	6	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗
US606	Process format (New)	58,33%	18	✗	✗	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗
US606a	Free-form format (New)	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US606b	Informal format (New)	50,00%	12	✗	✗	✓	✗	✓	✓	✗	✗	✓	✓	✓	✗
US606c	Structured format (New)	41,67%	6	✗	✗	✗	✓	✓	✓	✗	✗	✓	✗	✓	✗
US607	Artifact documentation (New)	16,67%	3	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗
US608	Multi-diciplinairy teams	25,00%	6	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗
US609	Continuous threat modeling (New)	41,67%	7	✗	✓	✗	✗	✗	✗	✓	✓	✗	✓	✓	✗
E007: Analysis															
US701	Resource valuation	8,33%	1	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
US702	Effect propagation	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
US703	Quality assurance	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US704	Threat library	25,00%	3	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
US705	Analysis level	41,67%	5	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗
US705a	Source code analysis	8,33%	1	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US705b	System design analysis	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
US705c	System architecture analysis	41,67%	5	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗
US706	Security objective	8,33%	1	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
US706a	Confidentiality	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US706b	Integrity	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US706c	Availability	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US706d	Privacy (New)	8,33%	1	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
US707	Prioritization	66,67%	8	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓
US708	Role of risk	58,33%	7	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓
US708a	Associate risk to identified threats	58,33%	7	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓

US708b	Consider risk externally	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US708c	Do not consider risk	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US709	Steps to vulnerability exploitation	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US710	Threat analysis type	16,67%	2	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
US710a	Qualitative threat analysis	16,67%	2	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
US710b	Quantitative threat analysis	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
E008: Other															
US801	Stopping condition / definition of done	16,67%	2	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗
US802	Interoperability	33,33%	10	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗
US803	Validation support	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
US804	Verification support	0,00%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US805	Automation	83,33%	36	✓✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓
US806	Portability	16,67%	4	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
US807	Marketplace (New)	8,33%	2	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
US808	Pipeline integration (New)	75,00%	21	✓	✓	✓	✗	✗	✓	✓	✓	✗	✓	✓	✓
US809	Flexible/usage (New)	25,00%	3	✗	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗
US810	Medium familiarity (New)	50,00%	19	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓
US810a	Tool familiarity (New)	25,00%	7	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
US810b	Technique familiarity (New)	50,00%	6	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓
US810c	Jargon (New)	25,00%	7	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✓
US811	Versioning (New)	16,67%	4	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
US812	Secure storage (New)	8,33%	1	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
US813	RBAC (New)	8,33%	1	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
US814	Step-by-step guidance (New)	33,33%	8	✗	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	✗
US815	Custom prioritization (New)	25,00%	5	✗	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗
US816	SDLC integration (New)	33,33%	8	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓	✓	✗
US817	Threat context (New)	25,00%	5	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	✗
US818	Classes and instances (New)	25,00%	3	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✓
US819	Templates (New)	25,00%	4	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗
US820	Collaboration (New)	8,33%	1	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
US821	Input data flexibility (New)	25,00%	4	✗	✗	✗	✗	✗	✗	✓	✓	✗	✓	✗	✗
US822	Threat modeling life cycle adminstration (New)	33,33%	4	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓
US823	Method consistency	8,33%	2	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
US824	Change visualization (New)	16,67%	2	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓
US825	Simple drawing tool	16,67%	2	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓
US826	Custom threat library (New)	41,67%	8	✓	✗	✗	✗	✗	✓	✗	✗	✓	✓	✓	✗
US827	Duplicate threat recognition (New)	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
US828	Model and code linkage (New)	8,33%	2	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
US829	Modification independence	8,33%	1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

C.2 Quality occurrence analysis

Table C.2: The considered qualities by the interviewees. Checkmarks denote the qualities mentioned by the domain experts, and cross marks signify that the domain expert did not mention the quality. The Mention rate denotes the unique mentions of the quality over all the interviews. The total mentions indicate the number of segments in which the quality has been detected.

Quality criteria	**Mention rate**	**Total mentions**	Interview 1	Interview 2	Interview 3	Interview 4	Interview 5	Interview 6	Interview 7	Interview 8	Interview 9	Interview 10	Interview 11	Interview 12
Applicability														
Accuracy (New)	66.67%	12	✓	✗	✗	✓	✓	✓	✓	✓	✓	✗	✓	✗
Clear process (New)	91.67%	34	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compatibility with agile development processes	33.33%	5	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗
Compatibility with related processes and tools	75%	25	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗
Organisational fit (New)	100%	40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Usefulness (New)	100%	45	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
User experience: Ease of use	91.67%	30	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
User experience: Learning curve	83.33%	21	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓
Efficiency														
Cost	83.33%	29	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓
Reusability	25.00%	3	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✓	✗
Scalability	16.67%	2	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
Tailorability	33.33%	7	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Uniformity (New)	16.67%	3	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗
Reliability														
Completeness	16.67%	2	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
Discoverability (New)	33.33%	7	✗	✗	✓	✗	✗	✓	✗	✓	✗	✓	✗	✗
Documentation: Technique	16.67%	2	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗
Documentation: Tool	16.67%	2	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
Effectiveness/Testability (New)	16.67%	2	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗
Maturity: Technique	16.67%	3	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗
Maturity: Tool	16.67%	3	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗
Repeatability	50%	10	✗	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	✗
Software evolution support: Change impact analysis	25%	4	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓
Software evolution support: Change propagation	16.67%	2	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
Software evolution support: Component architecture	0%	0	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Software evolution support: Modularity	25%	3	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗
Suitability	25%	5	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗
Support for maintainability: Modifiable	16.67%	4	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓
Support for maintainability: Traceability	33.33%	5	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓
Understandability	91.67%	19	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓

C.3 Quality x feature mapping

Table C.3: The full quality x requirement mapping. A 1 indicates that there is at least one co-occurrence was found between the quality and requirement and a 0 indicates there has not been any co-occurrence found. Some qualities have been shortened, due to readability. Requirements for which no highlight has been found are marked in red.

Quality x Requirement Mapping	**Applicability**									**Efficiency**					**Reliability**															
	ccuracy	lear process	omp. with Agile..	omp. with non-agile..	omp. with related..	rganisational fit	sefulness	ase of use	earning curve	ost	eusability	calability	ailorability	niformity	ompleteness	iscoverability	echnique documentation	ool documentation	ffectiveness	echnique maturity	ool maturity	epeatability	hange impact analysis	hange propagation	omponent architecture	odularity	uitability	odifiable	raceability	nderstandability
Actionable findings (US206)	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0	1
Analysis level (US705): System architecture analysis (c)	0	1	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Analysis level (US705): System design analysis (b)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Applicability to system (US503)	1	0	0	0	1	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
Approach (US301): Attack-centric (a)	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Approach (US301): Risk-centric (b)	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Approach (US301): Software-centric (c)	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Artifact documentation (US607)	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1
Automation (US805): Automatic effect propagation (d)	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1	0	1	1
Automation (US805): Automatic prioritization (c)	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	1	0	0	1	1	1	1	1	1	0	1	1
Automation (US805): Automatic system representation generation (b)	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1	0	0	1
Automation (US805): Automatic threat generation (a)	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	0	1	0	0	1	1	1	1	1	1	0	0	1
Change highlights (US824): Common control change (d)	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
Change highlights (US824): Organisational change (b)	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
Change highlights (US824): Prioritizatoin change (c)	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
Change highlights (US824): Threat input change (a)	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
Classes and instances (US818)	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Common vulnerabilities (US207)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Component templates (US819)	0	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0
Continuous threat modeling (US609)	1	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	0	0	0

Coverage of components (US406)	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0
Current security (US712)	0	0	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	1
Custom component library (US830)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Custom prioritization (US815)	0	0	0	0	1	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Custom threat library (US826)	1	0	0	0	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0
Deployment context (US506)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Effect propagation (US702)	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0
Flexible usage (US809)	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Focus (US302): Attacker view (a)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Focus (US302): Defender view (b)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hardware threats (US504)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Interoperability (US903)	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0
Involved entities (US602)	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Knowledge barriers (US501)	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Layer (US502): Environment level (d)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Layer (US502): Computer-based system level (b)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Layer (US502): Software level (a)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Layer (US502): Total system level (c)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Medium familiarity (US901): Jargon (c)	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Medium familiarity (US901): Technique familiarity (b)	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
Medium familiarity (US901): Tool familiarity (a)	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Modeling view (US304): Behavior view (b)	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Modeling view (US304): Platform view (c)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Modeling view (US304): Software architecture view (a)	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0
Multi-diciplinairy teams (US608)	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Multiple threat actors (US303)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Notation abstraction (US401): Goal-centric (a)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Notation abstraction (US401): Model-centric (b)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Notation abstraction (US401): Problem-centric (c)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Notation abstraction (US401): Process-centric (d)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Notation Extensibility (US405)	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0
Notation familiarity (US403)	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Notation Formality (US402)	0	1	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
Output granularity (US202): High-level outcome (a)	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Output granularity (US202): Low-level outcome (b)	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1
Output type (US201): Security requirements (c)	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Output type (US201): Security threats (a)	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Output type (US201): Threat mitigations (b)	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Pipeline integration (US808): Security tickets (a)	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	1	1
Pipeline integration (US808): Software development tracking (b)	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	1	1
Portability (US806)	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Prioritization (US707)	1	1	0	0	1	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
Priority visualization (US409)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Process format (US606): Free format (a)	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Process format (US606): Informal format (b)	0	1	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Process format (US606): Structured format (c)	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	0
Quality guidelines (US610)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Relationships between components (US407)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Report representation (US203): Figure (b)	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Report representation (US203): List (a)	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Resource valuation (US711)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Role of risk (US708): External risk (b)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Role of risk (US708): Include risk (a)	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
SDLC integration (US902)	0	0	1	0	1	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Security control visualization (US408)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Security objective (US706): Availability (c)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Security objective (US706): Confidentiality (a)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Security objective (US706): Integrity (b)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Security objective (US706): Privacy (d)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Security patterns (US205)	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Similarity with software specification languages (US404)	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Simple drawing tool (US825)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Step-by-step guidance (US814)	0	1	0	0	0	1	1	1	1	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1
Stopping condition (US611)	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
System decomposition (US505)	0	0	0	0	1	0	0	1	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	1	1	0	0	1	1
System input (US102): Architectural design (d)	1	1	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
System input (US102): Business process (f)	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
System input (US102): Descriptive language (g)	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
System input (US102): Security assumptions (b)	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
System input (US102): System goals (c)	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Threat analysis type (US710)	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Threat context (US817)	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Threat generation (US601): Knowledge / methodology based (a)	1	0	0	0	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
Threat generation (US601): User knowledge based (c)	1	1	0	0	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Threat input (US103): Brainstorming (a)	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Threat input (US103): Framework (i)	1	0	0	0	0	1	0	1	0	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
Threat input (US103): Historical data (b)	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Threat input (US103): Threat actors (h)	1	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0
Threat input (US103): Threat intelligence tooling (c)	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0
Threat input flexibility (US821)	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Threat library (US704)	1	0	0	0	1	0	0	1	1	1	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0
Threat modeling life cycle adminstration (US822)	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
Trigger (US605): Dynamic (b)	1	1	1	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
Trigger (US605): Static (a)	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trust boundaries (US831)	0	0	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0
Unique Threats (US604)	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Validation (US803)	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Verification (US804)	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Versioning (US811)	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0

D Interview artifacts

D.1 New requirements from interview

Table D.1: Additional requirements discovered during expert interviews. When a new sub-user story is discovered for an already existing main user story, the main user story is indicated with an Asterisk (*).

ID	Placeholder	User story - Description
E001	Input	
US102*	Input type	What type of input is required in order to start the threat analysis.
g	Descriptive language	As a decision-maker, I want the analysis to be done based-on a descriptive language, so it can be used to automatically generate diagrams. (E.g. YAML)
US103*	Input data	Where the input data comes from.
c	threat intelligence	As a decision-maker, I want the threat assessment to be done based-on threat intelligence tooling.
d	Detailed threats	As a decision-maker, I want the threat assessment to be done based-on detailed threats.
e	MITRE	As a decision-maker, I want the threat assessment to be done based-on MITRE ATT&CK framework.
f	Personas	As a decision-maker, I want the threat assessment to be done based-on personas.
g	Security cards	As a decision-maker, I want the threat assessment to be done based-on security cards.
E002	Output	
US205	Security patterns	As a decision-maker, I want the method to assist with identifying security patterns, so I can identify security areas that need to be fixed.
US206	Actionable findings	As a decision-maker, I want the outcome of the method to be actionable, so that the outcomes can be used in subsequent software development.
US207	Common vulnerabilities	As a decision-maker, I want the output of the method to identify threats related to the most common vulnerabilities. (E.g. from the OWASP top 10.)
E004	Modeling notation	

US409	Priority visualization	As a decision-maker I want to be able to visualize how important a threat is. (For example based on the CVSS score of the attached vulnerability)
E006	Method process	
US604	Unique threats	As a decision-maker, I want the method to identify threats that are unique in comparison with other security techniques and tools.
US605	Trigger	When the threat modeling method is initiated.
a	Static trigger	As a decision-maker, I want the threat modeling method to be built around static time slots, so threat modeling is done regularly.
b	Dynamic trigger	As a decision-maker, I want the threat modeling method to be built around dynamic sessions, so it will be done in a response to a change.
US606	Process format	The format of a process can either be free-form or structured.
a	Free format	As a decision-maker, I want the threat modeling method to be free-form, so users can decide on how to do the threat modeling.
b	Informal format	As a decision-maker, I want the threat modeling method to be informal, so users are free to act within certain boundaries. (E.g. do threat modeling in STRIDE)
c	Structured format	As a decision-maker, I want the threat modeling method to be strucutred so users are constraints be a formal process and or tool.
US607	Artifact documentation	As a decision-maker, I want the threat modeling method to have a documentated artifact.
US608	Multi-diciplinairy teams	As a decision-maker, I want the threat modeling method to have a documentated artifact.
US609	Continuous threat modeling	As a decision-maker, I want the threat modeling method to be continuous, so it is constantly updated beyond the initial development stages.
E008	Other	
US807	Marketplace	As a decision-maker, I want a tool to have a marketplace, so I can add plugins and add-ons to the tool.
US808	Pipeline inegration	As a decision-maker, I want the tool to be integrated into a pipeline, so it will be integrated with the other tools.
US809	Flexible usage	As a decision-maker, I want the tool to allow flexibility and not impose limitations on how threat modeling is performed.
US810	Medium familiarity	As a decision-maker, I want the method to match the context of the stakeholders, so they can easily use it.
a	Tool familiarity	As a decision-maker, I want the tool to match the context of the stakeholders, so they can easily use it. (E.g. command line, when users are developers)
b	Technique familiarity	As a decision-maker, I want the technique to match the context of the stakeholders, so they can easily use it. (E.g. use abuse cases, when familiar with use cases)
c	Medium familiarity	As a decision-maker, I want the jargon to match the context of the stakeholders, so they can understand the terms. (E.g. use simple terms for non-security users)
US814	Step-by-step guidance	As a decision-maker, I want the technique and tool to provide step-by-step guidance for the users, so the actions and their order is clear.

US815	Custom prioritization	As a decision-maker, I want to be able to configure the prioritization in the tool so I can indicate what threats are important based on organisation specific risk.
US816	SDLC integration	As a decision-maker, I want the method to be integrated in the software development lifecycle, so it's part of the development cycle.
US817	Threat context	As a decision-maker, I want the tool to provide indication of when threats are not applicable, so that false positives are quickly reduced.
US819	Templates	As a decision-maker, I want to be able to create templates in the tool, so they can be re-used.
US820	Collaboration	As a decision-maker, I want my team to be able to work together in the threat modeling tool.
US821	Input data flexibility	As a decision-maker, I want the tool to be able to import all kinds of threat related data, so threats from many different sources can be utilized.
US822	Threat modeling life cycle adminstration	As a decision-maker, I want the tool to administer the whole threat modeling cycle, so I can have consitency between information gathering and the threat analysis.
US823	Method consistency	As a decision-maker, I want the tool to make use of a cosistent method, so the underlying logic is not changed suddenly.
US824	Change visualization	As a decision-maker, I want the tool to visualize external changes that affect the system, so the impact of external changes can be explained.
US825	Simple drawing tool	As a decision-maker, I want the tool to be a simple drawing tool, so I can use it freely.
US826	Custom threat library	As a decision-maker, I want to be able to make a custom threat library, so I can identify threats relevant to what the organisation is trying to enforce. (includes compliancy with security standards)
US827	Duplicate threat recognition	As a decision-maker, I want the tool to indentify when threats are duplicate, so their mitigations can be disregarded. (E.g. if working with multiple teams)
US828	Model and code linkage	As a decision-maker, I want to be able to store the threat modeling projects together with the source code, so it's easier to sync when making modifications.
US829	Modification independence	As a decision-maker, I want to be able to modify my system representation outside of the tool interface.

D.2 Post-interview requirement change log

Table D.2: The changes made after the interview organised per epic.

ID	**Textual placeholder**	**Description**	**Change and reasoning**
E001	*Input*	*Everything that happens before starting the threat analysis. This regards components that are related to identifying and describing what is already in place in the organisation.*	No change.

~~**US101**~~	~~Current security~~	~~As a decision-maker, I want the method to take current security measures into account so that irrelevant threats can be disregarded. (E.g. DDOS threat disregarded due to DDOS protection implemented)~~	Moved to E007: analysis. Current security is an important requirement but as part of the prioritization process. You do not want to use it as input, because you do not want to start with a "Netto risk".
US102	~~Input type~~ System input	What type of input system representation is required in order to start the threat analysis.	The difference between input type and input data was not clear, so both are renamed.
~~US102a~~	~~Attacker behavior~~	~~As a decision-maker, I want the analysis to require an attacker behavior specification, so the analysis is based on what the system should be protected from.~~	Attacker behavior is combined with MITRE and personas into: "US103h: Threat attackers and behavior"
US102b	Security assumptions	As a decision-maker, I want the analysis to require security assumptions, so the analysis is based on what the system should be protected from.	No change.
US102c	System goals	As a decision-maker, I want the analysis to require system goals, so that the analysis can be performed as soon as a high-level description is provided.	No change.
US102d	Architectural design	As a decision-maker, I want the analysis to require the architectural design of the system, so that the analysis is performed based on a model of the system.	No change.
~~US102e~~	~~Source code~~	~~As a decision-maker, I want the analysis to be done based-on source code, so that the analysis is performed based on the actual system.~~	Removed due to low applicability. Experts mention there are other tools to do source code threat analysis.
US102f	Business processes	As a decision-maker, I want the analysis to be done based-on business processes, so that the user knows ~~the business context~~ the context in which the system operates.	Rephrased to make more clear.
US102g	Descriptive language	As a decision-maker, I want the analysis to be done based-on a descriptive language, so it can be used to automatically generate ~~diagrams~~ a system representation. (E.g. YAML)	Changed to be in line with the terminology used by practitioners.
US103	~~Input data~~ Threat input	Where the input data comes from. What data is used as basis for threat identification	The difference between input type and input data was not clear, so both are renamed.
US103a	~~Expert opinion~~ Brainstorming	As a decision-maker, I want the threat assessment be based-on ~~expert opinion~~ brainstorming, so that no additional input data is required.	Expert has more of a guiding role, so it is generalized to brainstorming.
US103b	Historical data	As a decision-maker, I want the threat identification to be done based-on historical data. (e.g. threats from previous threat models, or other activities which identify threats)	Example added for clarification.

US103c	Threat intelligence tooling	As a decision-maker, I want the threat assessment to be done based-on threat intelligence tooling.	No change.
~~US103d~~	~~Input data~~	~~As a decision-maker, I want the threat assessment to be done based-on detailed threats.~~	After further analysis, the segments marked with this threat discussed having a threat catalogue.
~~US103e~~	~~MITRE~~	~~As a decision-maker, I want the threat assessment to be done based-on MITRE ATT&CK framework.~~	Combined with personas and attacker behavior into: "US103h: Threat attackers and behavior"
~~US103f~~	~~Personas~~	~~As a decision-maker, I want the threat assessment to be done based-on personas.~~	Combined with MITRE and attacker behavior into: "US103h: Threat attackers and behavior"
~~US103g~~	~~Security cards~~	~~As a decision-maker, I want the threat assessment to be done based-on security cards~~	Removed due single mention.
US103h	Threat actors	As a decision-maker, I want the threat identification to be done based-on threat actors and their behavior*, so the identification is based on what the system should be protected from. (E.g. Use MITRE ATT&CK framework in combination with personas) **Threat actors and their behavior:** Expected actions and tactics used by a malicious actor to exploit vulnerabilities in computer systems.	Added as a combination of personas, MITRE and attacker behavior.
US103i	Framework	As a decision-maker, I want the threat identification to be done based-on a framework. (E.g. STRIDE)	This is not historical data but was classified as historical data or threat generation, during the qualitative coding. This requirement was added.
E002	*Output*	*Requirements related to the outcome of the threat modeling method.*	No change.
US201	Output type	The type of artifact that the method produces.	No change.
US201a	Security threats	As a decision-maker, I want the threat modeling method to produce relevant security threats, so I know which threats are applicable to my context.	Added refined description, included relevant due to one of the participants mentioning that threats themselves were the input.
US201b	Threat mitigations	As a decision-maker, I want the threat modeling method to produce threat mitigations, ~~so general risk-lowering countermeasures are obtained~~ so specific risk lowering measures are recommended.	Descriptions of security requirements and mitigations were mixed up, this change is a correction of the initial mistake.

US201c	Security requirements	As a decision-maker, I want the threat modeling method to produce security requirements, ~~so that a refined set of specific countermeasures is obtained~~ so that a set of rules that describe how systems can be protected against threats is obtained.	Descriptions of security requirements and mitigations were mixed up, this change is a correction of the initial mistake.
~~US201d~~	~~Security tickets~~	~~As a decision-maker, I want the threat modeling method to produce security tickets, so the development teams know what to fix.~~	Moved to E008: tool.
US202	Output granularity	The level of detail in which the result from threat modeling is presented.	No change.
US202a	High-level outcome	As a decision-maker, I want the threat modeling method to produce a high-level outcome, so a general overview can be obtained that can guide the action points. (An example of a high level outcome is a document that provides ~~an overview~~ a functional description of general security improvements.)	Example modified to be in line with the terminology used in the interviews.
US202b	Low-level outcome	As a decision-maker, I want the threat modeling method to produce a low-level outcome, so that the outcome can be directly used for implementation. (An example of a low-level outcome is a model of the improved system, which can directly be translated into code. technical description of each threat.)	Example modified to be in line with the terminology used in the interviews.
US203	Report representation	How the findings from threat modeling are displayed in the report.	No change.
US203a	~~Structured text~~ List	As a decision-maker, I want the report from a threat modeling project to be a structured text. (For example CSV) list. (For example, a list of applicable threats and their mitigations)	Participants were not familiar with the term structured text.
US203b	~~Model-based~~ Figure	As a decision-maker, I want the report from a threat modeling project to ~~have a model-based representation.~~ include a figure. (E.g. a model of the system and its threats or a graph of threats found)	Participants were not familiar with the term "model-based representation"
~~**US204**~~	~~Mitigation strategy~~	~~As a decision-maker, I want the method to assist in providing mitigation strategies.~~	Removed due to similar semantics as US201b: Threat mitigations.
US205	Security patterns	As a decision-maker, I want the method to assist with identifying security patterns, so I can identify security areas that need to be fixed.	No change.
US206	Actionable findings	As a decision-maker, I want the outcome of the method to be actionable, so that the outcomes can be used in subsequent software development.	No change.
US207	Common vulnerabilities	As a decision-maker, I want the output of the method to identify threats related to the most common vulnerabilities. (E.g. from the OWASP top 10.)	No change.
E003	*Perspective*	*Requirements related to the perspective of the threat modeling method. Contains components that represent the point of view from which the approach solves the threat modeling case*	No change.
US301	Approach	The central ideology that the threat modeling method is based on.	No change.
US301a	Attack-centric approach	As a decision-maker, I want the technique to be attack-centric, so there will be a focus on identifying attacker profiles and the complexity of attacks.	No change.
US301b	Risk-centric approach	As a decision-maker, I want the technique to be risk-centric, so that impact and likelihood of the threats decide which security requirement needs to be addressed first.	No change.

US301c	Software-centric approach	As a decision-maker, I want the technique to be software-centric, so the focus will be on the software that is examined. (E.g. STRIDE analysis on DFDs)	No change.
~~US301d~~	~~Approach~~	~~As a decision-maker, I want the technique to be centered around Goal-Oriented Requirement Engineering(GORE), so that both functional and non-functional (among other, security) requirements are obtained.~~	Remove requirement-based approaches, security specialists are not familiar with those. Additionally, they have a very low applicability rating.
~~US301e~~	~~Approach~~	~~As a decision-maker, I want the technique to be centered around Security Requirements Engineering (SRE), so that the primary goal is to identify security requirements.~~	Remove requirement-based approaches, security specialists are not familiar with those. Additionally, they have a very low applicability rating.
US302	Focus	The viewpoint from which the technique is performed.	No change.
US302a	Attacker focus	As a decision-maker, I want the technique to be designed from the point of view of the attacker.	No change.
US302b	Defender focus	As a decision-maker, I want the technique to be designed from the point of view of a defender.	No change.
US303	~~Agents~~ Multiple threat actors	~~Does a threat model focus on one perspective or multiple perspectives at the same time.~~ As a decision-maker, I want one model to show the perspectives of multiple threat actors so that conflicting perspectives can be considered. (For attack-centric approaches)	Because single agent is removed, the requirement has one sub-user story, so it is turned into a single requirement.
~~US303a~~	~~Agents~~	~~As a decision-maker, I want one model to show the perspectives of multiple agents so that conflicting perspectives can be considered. (For attack-centric approaches)~~	Removed due to reason mentioned above.
~~US303b~~	~~Agents~~	~~As a decision-maker, I want one model to show the perspective of a single agent, so that the model does not become too complicated. (For attack-centric approaches)~~	Single agent removed due to very low selection rate.
US304	Modeling view	The level of detail on which the ~~actual model is created~~ system is represented in the model.	Modified to be in line with the terminology used in the interviews.
US304a	Software architecture view	As a decision-maker, I want the threat model to describe the architecture of the system, so that threats can be related to system components. (This relates to both a single system and a chain of software systems)	Add further explanation, to make clear that it includes a chain of systems and that it is on the application layer.

US304b	Behavior view	As a decision-maker, I want the threat model to describe the ~~functional~~ behavior of the system. (E.g. Use state diagrams to do threat modeling) **Behavior:** The behavior of the system refers to the (expected) actions and outputs of the system. For example, a payment system transfers a balance from one account to the other, using the internet. When there is a DDOS attack on the system, this threat results in the function being denied. Meaning that the system is not able to transfer balance. (This includes functional behavior, as well as, purpose and process)	Removed functional to make it more widely applicable.
US304c	Platform view	As a decision-maker, I want the threat model to describe the platform of the system, so that operating systems, middleware and also physical parts are considered.	No change.
~~US304d~~	~~Process view~~	~~As a decision-maker, I want the threat model to describe the process surrounding the system.~~	Combined with functional behavior and turned into US304b: behavior.
E004	*Modeling notation*	*Requirements specific to the notation used to represent the threat model.*	No change.
US401	Notation Abstraction	The level of detail at which the components are represented in the notation.	No change.
US401a	Goal-centric	As a decision-maker, I want the model notation to represent its components centered around (security) goals.	No change.
US401b	Model-centric	As a decision-maker, I want the model notation to represent its components centered around a model of the system.	No change.
US401c	Problem-centric	As a decision-maker, I want the model notation to represent its components centered around (security) problems.	No change.
US401d	Process-centric	As a decision-maker, I want the model notation to represent its components centered around the business processes.	No change.
US402	Notation formality	As a decision-maker, I want the notation to be graphical, so that attack trees, attack graphs, Data Flow Diagrams (DFDs), or tables can be used.	Due to removing formal notations, one sub-user story is left, so it is turned into a single requirement.
~~US402a~~	~~Formality~~	~~As a decision-maker, I want the notation to be formal, so that threat modeling based on a mathematical model is possible.~~	Removed due to low applicability.
~~US402b~~	~~Formality~~	~~As a decision-maker, I want the notation to be graphical, so that attack trees, attack graphs, Data Flow Diagrams (DFDs), or tables can be used.~~	Moved to US402.
US403	~~Language~~ Notation familiarity	As a decision-maker, I want to use a notation for threat modeling that ~~I am familiar with. (E.g. DFD, petri-nets, UML)~~ the users are familiar with.	Familiarity is important only from the perspective of the user. This requirement is used in the decision-model as a global multiplier.
~~**US404**~~	~~Similarity with software specification languages~~	~~As a decision-maker, I want to use (an extension of) a notation that is a well-known software specification language, so that threat modeling becomes easier to learn.~~	Removed due to low applicability.
~~**US405**~~	~~Notation extensibility~~	~~As a decision-maker, I want to be able to extend the modeling notation, so that domain-specific components can be modeled.~~	Removed due to low applicability.

US406	Coverage of components	As a decision-maker, I want the notation to have a specific set of components present.	Relevant but complex benchmarking required. Not considered in the decision-model
US407	Relationships between components	As a decision-maker, I want to have a notation where the most common relationships between components are present.	Relevant but complex benchmarking required. Not considered in the decision-model
US408	~~Countermeasure impact~~ Security control visualization	As a decision-maker, I want to be able to visualize ~~how the proposed countermeasures affect the system's weaknesses, so I can understand how effective the mitigations are.~~ what security controls are implemented, so I can visualize the impact on the threats.	Countermeasure impact is interpreted as a benchmark of effectiveness, rather than being able to visually represent security controls. This caused the change to a more suitable terminology.
US409	Priority visualization	As a decision-maker I want to be able to visualize how important a threat is. (For example change color based on the CVSS score of the attached vulnerability)	No change.
E005	*Modeling context*	*These requirements are about the context in which the modeling method can or cannot be applied, which is called the scope of usage.*	No change.
US501	Barriers for practitioners	As a decision-maker, I want a threat modeling method that does not have any knowledge barriers.	No change. Relevant but can not be benchmarked accurately. This requirement is not considered in the decision-model.
US502	Layer	The level of system details at which threats are ~~assessed~~. identified.	Modified to be in line with the terminology used in the interviews.
US502a	Software layer	As a decision-maker, I want the threat modeling method to help identify threats on the software level, so that only software components are considered.	No change.
US502b	Computer-based layer	As a decision-maker, I want the threat modeling method to help identify threats on the computer-based system level, so that both software and hardware are considered.	No change.
US502c	Total system layer	As a decision-maker, I want the threat modeling method to help identify threats on the total system level, so that ~~man, technology and organisation (MTO) are considered.~~ so that people, process, and technology (PPT) are considered.	Modified to be in line with the terminology used in the interviews.
US502d	Environmental layer	As a decision-maker, I want the threat modeling method to help identify threats on the environment level, so that factors beyond ~~MTO~~ PPT are considered.	Modified to be in line with the terminology used in the interviews.

US503	Applicability to system	As a decision-maker, I want to have a threat modeling method that is generally applicable, so I can apply it on many different systems in my organisation	It was found that decision-makers wanted to have a general applicable system that was specific enough to identify threats differently for on-premise, cloud and hybrid systems. To deal with this, US503a becomes US503 and US503b becomes US506.
~~US503a~~	~~Applicability to system~~	~~As a decision-maker, I want to have a threat modeling method that is generally applicable, so I can use it in many different systems.~~	See above.
~~US503b~~	~~Applicability to system~~	~~As a decision-maker, I want to have a method that is specifically designed for one type of system. (An example of a specific method would be: a threat modeling method for only cloud applications)~~	See above.
US504	Hardware threats	As a decision-maker, I want to have a threat modeling method that considers threats to hardware. (E.g. physical tempering is considered)	No change.
US505	~~Layer~~ System decomposition	As a decision-maker, I want to have a threat modeling method that allows to switch between system ~~layers~~ abstractions, so I can match the system depth with the stakeholder.	Modified to be in line with the terminology used in the interviews.
US506	Deployment context	As a decision-maker, I want the threat modeling method to keep in mind the context in which the system is deployed, so identified threats are more relevant. (E.g. Cloud versus on premise)	Added to deal with the interpretation of US503a and US503b.
E006	*Method process*	*Requirements that apply to the process of threat modeling and its environment.*	No change.
US601	Threat generation	The process of identifying and establishing a list of potential threats.	No change.
US601a	Threat generation	As a decision-maker, I want to have a method that generates threats based on a knowledge base/methodology. (e.g. Threat identification using STRIDE)	No change.
~~US601b~~	~~Threat generation~~	~~As a decision-maker, I want to have a threat modeling method that generates threats based on expert knowledge. (e.g. CORAS)~~	Removed due to low applicability.
US601c	Threat generation	As a decision-maker, I want to have a threat modeling method that generates threats based on user knowledge.	No change.
US602	Involved entities	As a decision-maker, I want to have a threat modeling method that supports the participation of specific stakeholders. (e.g. Threat modeling by security specialist versus threat modeling by developer)	No change. Relevant but can not be benchmarked accurately. This requirement is not considered in the decision-model.

US603	Set boundary	As a decision-maker, I want the technique to clearly explain how to define the limits of the system, so that we can identify the area that needs protection. (E.g. only consider assets with a specific priority level)	Setting a boundary is found to be procedural and not something that should depend on a method. Could be supported in a tool by setting trust boundaries but has to be left to humans.
US604	Unique threats	As a decision-maker, I want the method to identify threats that are unique in comparison with other security techniques and tools.	No change. Relevant but can not be benchmarked accurately. This requirement is not considered in the decision-model.
US605	Trigger	When the threat modeling method is initiated.	No change. Relevant, but this requirement and its sub-user stories are too process dependent. These are not considered in the decision-model.
US605a	Dynamic	As a decision-maker, I want the threat modeling method to be built around static time slots, so threat modeling is done regularly.	No change. Not considered.
US605b	Static	As a decision-maker, I want the threat modeling method to be built around dynamic sessions, so it will be done in a response to a change.	No change. Not considered.
US606	Process format	The format of a process can either be free-form or structured.	No change.
US606a	Free format	As a decision-maker, I want the threat modeling method to be free-form, so users can decide on how to do the threat modeling.	No change.
US606b	Informal format	As a decision-maker, I want the threat modeling method to be informal, so users are free to act within certain boundaries. (E.g. do threat modeling in STRIDE)	No change.
US606c	Structured format	As a decision-maker, I want the threat modeling method to be structured so users are constraints be a formal process and or tool.	No change.
US607	Artifact documentation	As a decision-maker, I want the threat modeling method to have a documented artifact.	No change.
US608	Multi-disciplinary teams	As a decision-maker, I want the threat modeling method to have a documented artifact.	No change. Relevant but can not be benchmarked accurately. This requirement is not considered in the decision-model.
US609	Continuous threat modeling	As a decision-maker, I want the threat modeling method to be continuous, so it is constantly updated beyond the initial development stages.	No change.

US610	Quality guidelines	As a decision-maker, I want a technique that requires checking the quality of the model as part of the procedure. (e.g. Having guidelines for assessing the quality of the model)	Was originally requirement US703: Quality assessment.
US611	Stopping condition	As a decision-maker, I want to have a definition of done (stopping condition) in the technique, so users are uniformly guided on when the threat modeling is done.	Moved from epic: Other.
E007	*Analysis*	*Requirements that are related to the threat analysis. This mostly regards the steps involved, the reasoning used, and the aspects to be considered during the analysis."*	No change.
~~**US701**~~	~~Resource valuation~~	~~As a decision-maker, I want a threat modeling method that differentiates between critical and non-critical components.~~	Redefined to explicitly clarify this is done by stakeholders and moved to US711.
US702	Effect propagation	I want the threat modeling method to show how attacks on one part of the system can affect other parts that depend on it. (e.g. internet facing login application is compromised –>effect on user database)	No change.
US703	Quality assurance	I want a technique that ensures the quality of the outcome as part of the analysis procedure. (e.g. Use guidelines for assessing the quality of the model)	Moved and renamed to US610: Quality guidelines. This is done to be in line with the interpretation of the requirement.
US704	Threat library	As a decision-maker, I want a method that uses a list of known threats (a threat library) to identify applicable security threats based on the system's components.	No change.
US705	Analysis level	The level of detail used to analyze the system for potential threats.	No change. Requirement and its subcomponents are too multi-interpretable. Not considered in the decision-model.
~~US705a~~	~~Analysis level~~	~~As a decision-maker, I want the depth of the analysis to reach source code level.~~	Source code related requirements are removed.
US705b	System design analysis	As a decision-maker, I want the depth of the analysis to reach system design level, so it can produce a detailed outcome. (System design may include, technical details of security controls such as encryption standards, etc.)	No change. Not considered.
US705c	System architecture analysis	As a decision-maker, I want the depth of the analysis to reach architecture level, so it can be executed early on in the development process.	No change. Not considered.
US706	Security objective	The areas of security covered by the method.	No change.
US706a	Confidentiality	As a decision-maker, I want the security objective of the threat modeling method to cover confidentiality. (In other words, the threat modeling method can analyse threats to confidentiality)	No change.
US706b	Integrity	As a decision-maker, I want the security objective of the threat modeling method to cover integrity.	No change.

US706c	Availability	As a decision-maker, I want the security objective of the threat modeling method to cover availability.	No change.
US706d	Privacy	As a decision-maker, I want the security objective of the threat modeling method to cover privacy	No change.
US707	Prioritization	As a decision-maker, I want the method ~~to prioritize threats~~ to include a threat prioritization (done by stakeholders), so that we can determine the order in which to implement the countermeasures.	During the interview it was made clear that prioritization is important but is rather a human task.
US708	Risk	The role of risk in the threat modeling technique. Risk is expressed in terms of likelihood and impact of a malicious event.	No change.
US708a	Include risk	As a decision-maker, I want to have a threat modeling technique that associates risk (Likelihood & impact) to the identified threats on a specific component, so that threats can be prioritized	Added a component dimension. To be in line with the interviews.
US708b	External risk	As a decision-maker, I want to have a threat modeling technique that considers risk (Likelihood & impact) externally. (E.g. by combining the technique with ~~an external risk management framework~~ a risk management framework from the organisation)	Example changed to be in line with the interviews.
~~US708c~~	~~No risk~~	~~As a decision-maker, I want to have a threat modeling technique that does not consider risk. (E.g. because it is not needed to obtain the preferred threat modeling outcome)~~	Removed due to low applicability.
~~**US709**~~	~~Steps of vulnerability exploitation~~	~~As a decision-maker, I want to have a threat modeling method that considers steps to vulnerability exploitation (expressed in time) as part of the analysis.~~	Removed due to low relevance.
US710	Threat analysis type	~~The differentiation between quantitative and qualitative techniques for threat analysis.~~ As a decision-maker, I want my threat modeling analysis to be qualitative.	Due to removing quantitative threat modeling, one sub-user story is left, so it is turned into a single requirement.
~~US710a~~	~~Qualitative vs quantitative~~	~~As a decision-maker, I want my threat modeling analysis to be qualitative, so it is faster.~~	See above.
~~US710b~~	~~Qualitative vs quantitative~~	~~As a decision-maker, I want my threat modeling method to be quantitative, so it is more detailed.~~	Quantitative threat modeling is not considered. (Low applicability rating)
US711	Resource valuation	As a decision-maker, I want the stakeholders to be able to differentiates between critical and non-critical components, so the impact can be assesed based on the criticality of the assets.	Added. Was requirement US701.
US712	Current security	As a decision-maker, I want the method to take current security measures into account, so likelihood of the threats can be assessed in the appropriate context of the organisation. (E.g. DDOS threat disregarded/ranked at the bottom due to DDOS protection implemented)	Added. Was requirement US101.
E008	*~~Other~~ Tool*	*Requirements that ~~are not placed in a category, explicitly relate to the integration with the SDLC or relate to tool support.~~ relate to tools and automation.*	Other was changed to tool and moved to E009.

~~**US801**~~	~~Stopping condition~~	~~As a decision-maker, I want to have a definition of done (stopping condition) in the technique, so users are uniformly guided on when the threat modeling is done.~~	Moved to E006: Method process (US611).
~~**US802**~~	~~Interoperability~~	~~As a decision-maker, I want a threat modeling method that can work together with another cyber security method/tool. (E.g. Can the results of threat modeling be used in another security method?, or is the threat modeling method a part of a group of security methods?)~~	Moved to E009: Other (US903).
US803	Validation	As a decision-maker, I want to receive validation support for the threat model, so that I can be more sure that it accurately reflects the real-world situation. (e.g. Tool provides recommendations and feedback about common mistakes during the model development).	No change.
US804	Verification	As a decision-maker, I want to receive verification support for the threat model, so I can ~~guarantee the model is built correctly. (E.g. Only allow relationships between components when they are meaningful)~~ be more sure that the model is built correctly. (E.g. Highlight relationships between components when they are unlikely)	Change is made to be less restricting and provide a more accurate example.
US805	Automation	As a decision-maker, I want to have a tool that assists the threat modeling technique by (semi-)automating some of the steps.	Sub-user stories are added to account for the different types of automation a tool can have.
	Automatic threat generation	As a decision-maker, I want to have a tool that can automatically generate threats based on a threat library.	Added. See above.
	Automatic system representation generation	As a decision-maker, I want to have a tool that can automatically generate a system representation.	Added. See above.
	Automatic prioritization	As a decision-maker, I want to have a tool that can automatically prioritize threats. (Based on stakeholder input)	Added. See above.
	Automatic effect propagation	As a decision-maker, I want to have a tool that can do effect propagation automatically, so that threats on certain components affect those that depend on it.	Added. See above.
US806	Portability	As a decision-maker, I want a tool that can ~~transfer projects from one device to another. (For example, through an export and import function.)~~ import and export system diagrams.	Modified to be in line with the terminology used in the interviews.
~~**US807**~~	~~Marketplace~~	~~As a decision-maker, I want a tool to have a marketplace, so I can add plugins and add-ons to the tool.~~	Removed due to lack of support, only one mention.
US808	Pipeline integration	As a decision-maker, I want the tool to be that can be integrated into a pipeline, so it will be integrated with the other tools.	Rephrased to be less restricting.
US809	Flexible usage	As a decision-maker, I want the tool to allow flexibility and not impose limitations on how threat modeling is performed.	No change. This requirement is not considered in the decision-model due to inaccurate benchmarking.
~~**US810**~~	~~Medium familiarity~~	~~As a decision-maker, I want the method to match the context of the stakeholders, so they can easily use it.~~	Moved to E009: Other (US901).

~~US810a~~	~~Medium familiarity~~	~~As a decision-maker, I want the tool to match the context of the stakeholders, so they can easily use it. (E.g. command line, when users are developers)~~	Moved to E009: Other (US901a).
~~US810b~~	~~Medium familiarity~~	~~As a decision-maker, I want the technique to match the context of the stakeholders, so they can easily use it. (E.g. use abuse cases, when familiar with use cases)~~	Moved to E009: Other (US901b).
~~US810c~~	~~Medium familiarity~~	~~As a decision-maker, I want the jargon to match the context of the stakeholders, so they can understand the terms. (E.g. use simple terms for non-security users)~~	Moved to E009: Other (US901c).
US811	Versioning	As a decision-maker, I want the tool to have the capability to maintain a record of the versions of threat models, allowing me to track the changes made over time.	No change.
~~**US812**~~	~~Secure storage~~	~~As a decision-maker, I want the tool to store the threat models in a secure storage, so confidentiality is insured.~~	Removed due to lack of support, only one mention.
~~**US813**~~	~~RBAC~~	~~As a decision-maker, I want the access to the threat models to be restricted by RBAC, so that only the appropriate stakeholders can view and edit them.~~	Removed due to lack of support, only one mention.
US814	Step-by-step guidance	As a decision-maker, ~~I want the technique and tool to provide step-by-step guidance for the users, so the actions and their order is clear.~~ I want the tool to provide step-by-step guidance for the users, so the actions and their order is clear.	Switched from technique applicability to tool, since it is observed to be more important in the context of a tool.
US815	Custom prioritization	As a decision-maker, I want to be able to configure the prioritization in the tool so I can indicate what threats are important based on context and corresponding organisation specific risk.	No change.
~~**US816**~~	~~SDLC integration~~	~~As a decision-maker, I want the method to be integrated in the software development lifecycle, so it's part of the development cycle.~~	Moved to E009: Other (US902).
US817	Threat context	As a decision-maker, I want the tool to provide indication of when threats are not applicable, so that false positives are quickly reduced.	No change.
US818	Classes and instances	As a decision-maker, I want the tool to distinguish between classes and instances of components, so customized threat assessments based on each specific instance	No change.
US819	Component templates	As a decision-maker, I want to be able to ~~create templates in the tool, so they can be re-used.~~ have templates in the tool, so I can simply drag and drop components.	Modified to be in line with the context mentioned in the interviews.
~~**US820**~~	~~Collaboration~~	~~As a decision-maker, I want my team to be able to work together in the threat modeling tool.~~	Removed due to lack of support, only one mention.
US821	Input data flexibility	As a decision-maker, I want the tool to be able to import all kinds of threat related data, so threats from many different sources can be utilized.	No change.
US822	Threat modeling life cycle administration	As a decision-maker, I want the tool to administer the whole threat modeling cycle, so I can have consistency between information gathering and the threat analysis. (This includes the risk assessment & threat identification withing one tool)	Added extra clarification to clarify the scope.

~~**US823**~~	~~Method consistency~~	~~As a decision-maker, I want the tool to make use of a consistent method, so the underlying logic is not changed suddenly.~~	Removed due to lack of support, only one mention.
US824	Change ~~visualization~~ highlights	As a decision-maker, I want the tool to ~~visualize external changes that affect the system, so the impact of external changes can be explained.~~ highlight changes external to the system, so the impact of these changes can be assessed on the system.	Modified to be in line with the context mentioned in the interviews. Four sub-user stories are added based on the input during the interviews.
US824a	Threat input change	As a decision-maker, I want the tool to highlight changes to the threat input, so the user knows why new threats occur.	Added. See above.
US824b	Organisational change	As a decision-maker, I want the tool to highlight organisational changes that might affect the threat model, so that the user knows the threat model should be updated. (E.g. change in regulations or laws that he organisation must oblige to)	Added. See above.
US824c	Prioritization change	As a decision-maker, I want the tool to highlight changes in the threat radar, so that the user knows that the prioritization should be updated.	Added. See above.
US824d	Common control change	As a decision-maker, I want the tool to highlight changes in common controls, so the threat model should be updated. (E.g. MFA is enrolled organisation wide, some threats can therefore be accepted)	Added. See above.
US825	Simple drawing tool	As a decision-maker, I want the tool to be a simple drawing tool, so I can use it freely.	No change.
US826	Custom threat library	As a decision-maker, I want to be able to make a custom threat library, so I can identify threats relevant to what the organisation is trying to enforce. (Includes threats to compliancy with security standards)	No change.
~~**US827**~~	~~Duplicate threat recognition~~	~~As a decision-maker, I want the tool to identify when threats are duplicate, so their mitigations can be disregarded. (E.g. if working with multiple teams)~~	Removed due to lack of support, only one mention.
~~**US828**~~	~~Model and code linkage~~	~~As a decision-maker, I want to be able to store the threat modeling projects together with the source code, so it's easier to sync when making modifications.~~	Removed due to lack of support, only one mention. Can be enforced outside of the method.
~~**US829**~~	~~Modification independence~~	~~As a decision-maker, I want to be able to modify my system representation outside of the tool interface.~~	Removed due to lack of support, only one mention.
US830	Custom component library	As a decision-maker, I want to be able to make a custom component library, so I can add components specific to my organisation. (E.g. Can be used in combination with a custom threat library to identify domain specific threats.	Added based on discussions about notation extensibility (multi-interpretable).
US831	Trust boundaries	As a decision-maker, I want the users to be able to set trust boundaries and zones, so that we can identify the areas that need protection.	Tool instance of set boundary removed requirement (US603). Here, the user input is highlighted.

E009	*Other*	*Requirements that are not placed in a category or explicitly relate to the integration within the SDLC / organisation. These are seen as very context dependent and are therefore hard to benchmark.*	Originally E008. Now, this does not include tools.
US901	Medium familiarity	As a decision-maker, I want the method to match the context of the stakeholders, so they can easily use it.	No change.
	Medium familiarity	As a decision-maker, I want the tool to match the context of the stakeholders, so they can easily use it. (E.g. command line, when users are developers)	No change. Not considered. Can be future work.
	Medium familiarity	As a decision-maker, I want the technique to match the context of the stakeholders, so they can easily use it. (E.g. use abuse cases, when familiar with use cases)	No change. This requirement is used in the decision-model as a global multiplier.
	Medium familiarity	As a decision-maker, I want the jargon to match the context of the stakeholders, so they can understand the terms. (E.g. use simple terms for non-security users)	No change. Not considered. Can be future work.
US902	SDLC integration	As a decision-maker, I want the method to be integrated in the software development lifecycle, so it's part of the development cycle.	No change.
US903	Interoperabilit	As a decision-maker, I want a threat modeling method that can work together with another cyber security method/tool. (E.g. Can the results of threat modeling be used in another security method? or Are the outcomes of threat monitoring added to the threat library?)	No change.

D.3 Qualities after interviews

Table D.3: The refined quality hierarchy as-is after the interview phase. Note that some definitions have been reformulated to fit the context of threat modeling method selection. All qualities can be assessed on a three-level scale (high, medium, or low).

Criteria group	**Quality criteria**	**Quality sub-criteria**	**Description**	**Origin**
Efficiency			The method can be performed at minimal cost and effort	[19]
	Cost		The amount of resources to produce a threat model. Both the cost for creating a threat model (time/effort) and the cost for the purchase of a threat modeling tool are considered.	[159]
	Reusability		The ability to use parts of one threat model to create a new model.	[34, 43]
	Tailorability		Capability of a method to adapt to a specific appliance.	[156]
	Scalability		If a method is applicable to large systems or not. There are components in the method that scale with the size and complexity of the system under assessment.	[140, 159]
	Uniformity		The threat modeling method is consistent, standardized, or homogeneous across different teams or departments.	Interview
Reliability			The method is semantically correct and meaningful	[19]

	Completeness		Does the method specify a state or condition of having the necessary or appropriate parts of a threat model.	[34, 89, 90, 110, 111, 117, 150]
	Accuracy		Does the method provide results that correctly represent reality	Interview
	Repeatability		How consistent the results of threat modeling are when conditions are the same.	[90]
	Effectiveness / testability		Are the outcomes of the method implemented and effective in treating the threat.	Interview and Survey
	Documentation		The documenation of the method is clear and extensive.	[156, 159]
		Technique documentation	The technique is documented clearly and extensively.	[156, 159]
		Tool documentation	The tool has a clear and extensive documentation.	[156, 159]
	Software evolution support		How well a threat modeling method supports the continuous change and improvement of software throughout its lifecycle.	[34]
		Modularity	How easy is it to develop and use software components separately.	[34]
		Component architecture	The level of support for a component-based structure that allows software modules to be added and removed with ease.	[34]
		Change propagation	Ability to keep track of changes made to the system and to as well as the external world guarantee that a change is correctly propagated such that no inconsistent dependency is left unresolved.	[34]
		Change impact analysis	Ability to evaluate the effect that changes made to specific artefacts will have on other system components.	[34]
	Suitability		The aim and outcome of the method is inline with the expectations of the stakeholders.	[29, 156]
	Maturity		How well the different parts of the method are structured and designed.	[110, 156, 159]
		Technique maturity	The technique supported by a systematic and structured process.	[110, 140, 156, 159]
		Tool maturity	Degree of optimization of the tool for standard usage. (Values based on the capability maturity model)	[110, 156, 159]
	Support for maintainability		How easy it is to make and track changes while using the method.	[89]
		Modifiable	The ability to make changes to an existing threat model.	[89]
		Traceability	Concerns the ability to trace the history, application and location of threat modeling components.	[34, 89, 117]
		Discoverability	The ability to discover the results from threat modeling. (e.g. How do people know that the team has done threat modelling? How do people know where to find the results?)	Interview and Survey
	Understandability		How easy is it is to understand the outcome of the method.	[34, 90, 117]
Applicability			The users are able to apply the method	[19]

	User experience		The experience of a person who uses a threat modeling method.	[22]
		Ease of use	Simplicity of use of the technique, tool and notation.	[89, 90, 110, 111, 150, 156, 159]
		Learning curve	The cost associated with learning the threat modeling technique, tool and notation.	[156, 159]
	Organisational fit		Compatibility or alignment between the threat modeling method and the organisation in terms of goals, culture, and employee skills.	Interview
	Compatibility with development process		The technique works well in a (agile) development environment.	[159]
	Compatibility with related processes & tools		How well the threat modeling method works with other (security) related processes such as risk assessment. Is the overlap between other processes minimised?	[189]
	Clear process		Intuitiveness of the threat modelling method for the stakeholders intended to apply it.	Interview and Survey
	Usefulness		The capacity of the threat modeling method to be beneficial, valuable, or helpful in achieving the desired purpose or goal.	Interview

E Decision model artifacts

E.1 Relevancy assessment

Table E.1: The complete relevancy assessment of the refined requirements. Checkmarks indicate that the requirement is assessed based on the component. Cross marks signify the component is not relevant for benchmarking the requirement. An asterisk (*) indicates that a requirement is still relevant but not used as part of the current decision-model. A double asterisk (**) indicates the requirement is used in the multiplier. The description of each requirement can be found in Appendix D.2.

ID	Name	Tool	Tech.	Notation	Process
	E001: Input				
US102	**System input**	No assessment needed			
b	Security assumptions	✗	✓	✗	✓
c	System goals	✗	✓	✗	✓
d	Architectural design	✗	✓	✗	✓
f	Business processes	✗	✓	✗	✓
g	Descriptive language	✓	✗	✗	✗
US103	**Threat input**	No assessment needed			
a	Brainstorming	✗	✓	✗	✓
b	Historical data	✗	✓	✗	✓
c	Threat intelligence tooling	✗	✓	✗	✓
h	Threat actors	✗	✓	✗	✓
g	Framework	✗	✓	✗	✗
	E002: Output				
US201	**Output type**	No assessment needed			
a	Security threats	✗	✓	✗	✗
b	Threat mitigations	✗	✓	✗	✗
c	Security requirements	✗	✓	✗	✗
US202	**Output granularity**	No assessment needed			
a	High-level granularity	✓	✓	✗	✓
b	Low-level granularity	✓	✓	✗	✓
US203	**Report representation**	No assessment needed			
a	List	✓	✓	✗	✗

b	Figure	✓	✓	✗	✓
US205	Security patterns	✓	✓	✗	✓
US206	Actionable findings	✓	✓	✗	✓
US207	Common vulnerabilities	✓	✓	✗	✓
	E003: Perspective				
US301	**Approach**	No assessment needed			
a	Attack-centric approach	✗	✓	✗	✗
b	Risk-centric approach	✗	✓	✗	✗
c	Software-centric approach	✗	✓	✗	✗
US302	**Focus**	No assessment needed			
a	Attacker focus	✗	✓	✗	✗
b	Defender focus	✗	✓	✗	✗
US303	Multiple threat actors	✗	✓	✗	✗
US304	**Modeling view**	No assessment needed			
a	Software architecture view	✗	✓	✓	✗
b	Behavior view	✗	✓	✓	✗
c	Platform view	✗	✓	✓	✗
	E004: Modeling notation				
US401	**Notation Abstraction**	No assessment needed			
a	Goal-centric	✗	✗	✓	✗
b	Model-centric	✗	✗	✓	✗
c	Problem-centric	✗	✗	✓	✗
d	Process-centric	✗	✗	✓	✗
US402	Notation formality	✗	✗	✓	✗
US403	Notation familiarity**	✗	✗	✓	✗
US404	Similarity with software specification languages	✗	✗	✓	✗
US405	Notation extensibility	✗	✗	✓	✗
US406	Coverage of components*	✗	✗	✓	✗
US407	Relationships between components*	✗	✗	✓	✗
US408	Security control visualization	✗	✗	✓	✗
US409	Priority visualization	✗	✗	✓	✗
	E005: Modeling context				
US501	Knowledge barriers*	✓	✓	✓	✗
US502	**Layer**	No assessment needed			
a	Software layer	✗	✓	✗	✗
b	Computer-based layer	✗	✓	✗	✗
c	Total system layer	✗	✓	✗	✗
d	Environmental layer	✗	✓	✗	✗
US503	Applicability to system	✗	✓	✗	✗
US504	Hardware threats	✗	✓	✗	✗
US505	System decomposition	✓	✓	✗	✗
US506	Deployment context	✓	✓	✓	✓
	E006: Method process				
US601	**Threat generation**	No assessment needed			
a	Methodology-based threat generation	✗	✓	✗	✓
c	User-based threat generation	✗	✓	✗	✓
US602	Involved entities*	✗	✓	✗	✓
US604	Unique threats*	✗	✓	✗	✓

US605	**Trigger**	No assessment needed			
a	Dynamic*	✗	✗	✗	✓
b	Static*	✗	✗	✗	✓
US606	**Process format**	No assessment needed			
a	Free format	✗	✓	✗	✓
b	Informal format	✗	✓	✗	✓
c	Structured format	✗	✓	✗	✓
US607	Artifact documentation	✗	✓	✗	✓
US608	Multi-diciplinairy teams	✗	✗	✗	✓
US609	Continuous threat modeling	✗	✓	✗	✓
US610	Quality guidelines	✗	✓	✗	✓
US611	Stopping condition	✗	✗	✗	✓
E007: Analysis					
US702	Effect propagation	✗	✓	✗	✓
US704	Threat library	✗	✓	✗	✓
US705	**Analysis level**	No assessment needed			
b	System design analysis*	✗	✓	✗	✗
c	System architecture analysis*	✗	✓	✗	✗
US706	**Security objective**	No assessment needed			
a	Confidentiality	✗	✓	✗	✗
b	Itegrity	✗	✓	✗	✗
c	Availability	✗	✓	✗	✗
d	Privacy	✗	✓	✗	✗
US707	Prioritization	✓	✓	✗	✓
US708	**Risk**	No assessment needed			
a	Include risk	✓	✓	✗	✓
b	External risk	✓	✓	✗	✓
US710	Threat analysis type	✗	✓	✗	✓
US711	Resource valuation	✓	✓	✗	✓
US712	Current security	✓	✓	✗	✓
E008: Tools					
US803	Validation	✓	✗	✗	✗
US804	Verification	✓	✗	✗	✗
US805	**Automation**	No assessment needed			
a	Automatic threat generation	✓	✗	✗	✗
b	Automatic system representation generation	✓	✗	✗	✗
c	Automatic prioritization	✓	✗	✗	✗
d	Automatic effect propagation	✓	✗	✗	✗
US806	Portability	✓	✗	✗	✗
US808	**Pipeline integration**	No assessment needed			
a	Security tickets	✓	✗	✗	✗
b	Software development tracking	✓	✗	✗	✗
US809	Flexible usage	✓	✗	✗	✗
US811	Versioning	✓	✗	✗	✗
US814	Step-by-step guidance	✓	✗	✗	✗
US815	Custom prioritization	✓	✗	✗	✗
US817	Threat context	✓	✗	✗	✗
US818	Classes and instances	✓	✗	✗	✗

US819	Component templates	✓	✗	✗	✗
US821	Input data flexibility	✓	✗	✗	✗
US822	Threat modeling life cycle adminstration	✓	✗	✗	✗
US824	**Change highlights**	No assessment needed			
a	Threat input change	✓	✗	✗	✗
b	Organisational change	✓	✗	✗	✗
c	Prioritization change	✓	✗	✗	✗
d	Common control change	✓	✗	✗	✗
US825	Simple drawing tool	✓	✗	✗	✗
US826	Custom threat library	✓	✗	✗	✗
US830	Custom component library	✓	✗	✗	✗
US831	Trust boundaries	✓	✗	✗	✗
E009: Other					
US901	**Medium familiarity**	No assessment needed			
a	Medium familiarity*	✓	✗	✗	✓
b	Medium familiarity**	✗	✓	✗	✓
c	Medium familiarity*	✗	✓	✗	✓
US902	SDLC integration	✓	✓	✗	✓
US903	Interoperability	✓	✓	✗	✓

E.2 Requirement x Alternative mapping

Table E.2: The requirement and alternative mapping. A zero indicates that there has been no evidence that the requirement is met by the alternative. A 0.5 indicates partial supportability. A one indicates that the requirement is met by the alternative. A double asterisk(**) indicates the requirement is used as a global multiplier.

ID	Placeholder	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18
E001: Input																			
US102	**System input**	**No benchmark needed**																	
b	Security assumptions	1	1	1	1	1	0,5	1	0,5	0,5	0,5	0	1	1	1	0,5	0,5	0,5	0,5
c	System goals	0,5	0,5	0,5	0,5	0,5	0,5	0,5	1	1	0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0
d	Architectural design	1	1	1	1	1	0,5	1	0,5	0,5	0	1	1	1	1	0,5	0,5	0,5	0
f	Business processes	0,5	0,5	0,5	0,5	0,5	1	0,5	1	1	0,5	1	0,5	0,5	1	1	1	1	0,5
g	Descriptive language	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
US103	**Threat input**	**No benchmark needed**																	
a	Brainstorming	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
b	Historical data	0,5	0,5	1	0,5	1	1	0,5	0,5	0,5	1	1	0,5	1	1	1	1	1	1
c	Threat intelligence tooling	0,5	0,5	1	0,5	1	1	0,5	0,5	0,5	1	1	0,5	1	1	1	1	1	1
h	Threat actors	0,5	0,5	1	0,5	1	1	0,5	0,5	0,5	1	1	0,5	1	1	1	1	1	1
g	Framework	1	1	1	1	1	0	1	0	0	0	0	1	1	1	0	0	0	1
E002: Output																			
US201	**Output type**	**No benchmark needed**																	

a	Security threats	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
b	Threat mitigations	1	1	1	1	1	0,5	1	0,5	0,5	0,5	1	1	1	0,5	0,5	0,5	0,5	0,5
c	Security requirements	0,5	0,5	0,5	0,5	0,5	1	0,5	0,5	0,5	0,5	0,5	0,5	0,5	1	0,5	0,5	0,5	0,5
US202	**Output granularity**	**No benchmark needed**																	
a	High-level granularity	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
b	Low-level granularity	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0
US203	**Report representation**	**No benchmark needed**																	
a	List	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
b	Figure	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
US205	Security patterns	1	1	1	1	1	1	1	0	0	1	0	1	1	0	1	0	0	1
US206	Actionable findings	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
US207	Common vulnerabilities	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0
E003: Perspective																			
US301	**Approach**	**No benchmark needed**																	
a	Attack-centric approach	0	0	0	0	0	0,5	0	1	1	1	0	0	0	0	0	0	0	1
b	Risk-centric approach	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0
c	Software-centric approach	1	1	1	1	1	0,5	1	0	0	0	0	1	1	0	1	0	1	0
US302	**Focus**	**No benchmark needed**																	
a	Attacker focus	0,5	0,5	0,5	0,5	0,5	1	0,5	1	1	1	0,5	0,5	0,5	0,5	1	0,5	0,5	1
b	Defender focus	1	1	0,5	1	0,5	0,5	1	1	1	0	0,5	1	0,5	0,5	0	0,5	1	0
US303	Multiple threat actors	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
US304	**Modeling view**	**No benchmark needed**																	
a	Software architecture view	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1
b	Behavior view	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1
c	Platform view	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1
E004: Modeling notation																			
US401	**Notation Abstraction**	**No benchmark needed**																	
a	Goal-centric	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	1
b	Model-centric	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	0	0	1
c	Problem-centric	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1
d	Process-centric	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1
US402	Notation formality	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	1	0
US403	Notation familiarity**	**No benchmark needed**																	
US404	Similarity with software specification languages	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
US405	Notation extensibility	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	0	0	1
US408	Security control visualization	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0
US409	Priority visualization	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
E005: Modeling context																			
US502	**Layer**	**No benchmark needed**																	
a	Software layer	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
b	Computer-based layer	0	0	0	0	0	1	0	1	1	1	1	0	0	1	1	1	1	1
c	Total system layer	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	1	1
d	Environmental layer	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1
US503	Applicability to system	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
US504	Hardware threats	0	0	0	0	0	1	0	1	1	1	1	0	0	1	1	1	1	1
US505	System decomposition	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
US506	Deployment context	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

E006: Method process																			
US601	**Threat generation**	**No benchmark needed**																	
a	Methodology-based threat generation	1	1	1	1	1	0	1	0	0	0	0	1	1	1	0	0	0	1
c	User-based threat generation	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
US606	**Process format**	**No benchmark needed**																	
a	Free format	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
b	Informal format	1	1	1	1	1	0	1	0,5	0,5	0,5	0,5	1	1	0,5	0,5	0,5	0,5	1
c	Structured format	1	1	1	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0
US607	Artifact documentation	1	1	1	1	1	0,5	1	0,5	0,5	0,5	1	1	1	1	0,5	0,5	0,5	0,5
US609	Continuous threat modeling	0,5	0,5	0,5	0,5	0,5	1	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
US610	Quality guidelines	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	1	0,5	0,5	1	0,5	0,5	0,5	0,5
US611	Stopping condition	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E007: Analysis																			
US702	Effect propagation	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
US704	Threat library	1	1	1	1	1	1	1	0,5	0,5	0,5	0,5	1	1	0,5	0,5	0,5	0,5	1
US706	**Security objective**	**No benchmark needed**																	
a	Confidentiality	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
b	Itegrity	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
c	Availability	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
d	Privacy	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
US707	Prioritization	0,5	0,5	0,5	1	1	1	0,5	0,5	0,5	0,5	1	0,5	0,5	1	0,5	1	0,5	0,5
US708	**Risk**	**No benchmark needed**																	
a	Include risk	0	0	0	1	1	1	1	0	0	0	1	0	0	1	0	1	0	0
b	External risk	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
US710	Threat analysis type	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
US711	Resource valuation	0,5	0,5	0,5	1	1	1	1	0,5	0,5	0,5	1	0,5	0,5	1	0,5	0,5	0,5	0,5
US712	Current security	1	1	1	1	1	1	1	0,5	0,5	0,5	1	1	1	0,5	0,5	0,5	0,5	0,5
E008: Tools																			
US803	Validation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
US804	Verification	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
US805	**Automation**	**No benchmark needed**																	
a	Automatic threat generation	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
b	Automatic system representation generation	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
c	Automatic prioritization	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
d	Automatic effect propagation	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
US806	Portability	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
US808	**Pipeline inegration**	**No benchmark needed**																	
a	Security tickets	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
b	Software development tracking	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
US811	Versioning	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
US814	Step-by-step guidance	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
US815	Custom prioritization	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
US817	Threat context	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
US818	Classes and instances	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
US819	Component templates	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
US821	Input data flexibility	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

US822	Threat modeling life cycle adminstration	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
US824	**Change highlights**	**No benchmark needed**																	
a	Threat input change	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
b	Organisational change	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	Prioritization change	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	Common control change	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
US825	Simple drawing tool	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1	0
US826	Custom threat library	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
US830	Custom component library	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
US831	Trust boundaries	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E009: Other																			
US901	**Medium familiarity**	**No benchmark needed**																	
b	Medium familiarity**	**No benchmark needed**																	
US902	SDLC integration	1	1	1	1	1	1	1	0,5	0,5	0,5	0,5	1	0,5	0,5	0,5	0,5	0,5	0,5
US903	Interoperability	0,5	0,5	0,5	1	1	1	0,5	0,5	0,5	1	1	0,5	0,5	1	0,5	0,5	0,5	1

F Full survey

The last pages of this report will contain the survey as presented to the domain experts.

Interview preparation survey

Start of Block: Introduction

Thank you for agreeing to participate in an interview regarding the selection problem of threat modeling methods. A threat modeling method is a process that can be used to analyze potential attacks or threats. Given these threats, appropriate mitigation techniques are picked. There are numerous threat modeling methods, each with its own strengths and weaknesses. This makes it challenging for practitioners to choose a method that suits their situation. The goal of this research is to simplify this problem by providing a structured method that guides the selection. To accomplish this, we have gathered a comprehensive collection of threat modeling methods, quality criteria, and requirements from literature. These will serve as the foundation for a multi-criteria decision-making framework, which will assist practitioners in navigating through this big cluster of threat modeling methods.

To make this research useful in real-world settings, we need your help with refining the concepts found in literature. We are convinced that your experience will highly increase the quality of the research. To help streamline the interview process, we have prepared a survey to fill out **beforehand**. We are very delighted that you are willing to help and we greatly appreciate your participation.

You are asked to answer the questions from the point of view of a **decision-maker** responsible for selecting a threat modeling method for a development project. The survey consists of three parts. First we explore how you currently choose a threat modeling method. Then we will introduce you to the quality criteria and hierarchical structure. Afterward we will go in-depth on the requirements. During the interview, we will discuss your answers so we can refine these concepts and make them practical. At the end of each section, there will be a text field where you can ask any questions or provide comments. The survey can be completed at your own pace, and you can come back to it if needed. The duration of the survey is expected to be 30-45 minutes. I have sent you a personalized link that you can use to access your answers at any time. Meaning, you can take a break whenever you prefer. Thanks again for your participation!

Answers given in the survey may be questioned during the actual interview. The answers provided will not be used in our research unless consent is provided at the start of the upcoming interview.

Thanks, Lennard Marck

End of Block: Introduction

Start of Block: Part 0: Exploration

Firstly, we will ask you a question about how you currently choose a **threat modeling method***. You can give us as much information as you like. Company specific details will be removed after analysis. During the interview, we will go into more detail and ask additional questions.

A **threat modeling method** is a process that can be used to analyze potential attacks or threats. Given these threats, appropriate mitigation strategies are picked. In the context of this research, a method contains a technique, tool and notation.
A **technique** is a stepwise procedure, possibly with a prescribed notation, to perform a development activity. For example, a simple technique could include three steps: drawing system architecture, identifying threats and mitigating threats.
A **tool** is a possibly automated means to support a part of the development process. For instance, Microsoft TMT is a tool that can be used to draw a system architecture and identify threats.
A **notation** is a system of symbols with a corresponding set of rules to construct artifacts used in communication. An example of a notation is UML.

Name Please enter your name

____

0.1.1 If you would be responsible for threat modeling in your organization, how would you currently select a threat modeling method?

____

____

____

____

____

End of Block: Part 0: Exploration

Start of Block: Part 1: Quality criteria

This part is about putting quality criteria in order of importance, based on the perspective of a **decision-maker** responsible for selecting a threat modeling method for a development project.

A quality is a special attribute or characteristic that can be applied to a method. In the context of this research, a quality is not directly measurable. An example could be: ease of use of the threat modeling method. In order to represent the degree of quality satisfaction, they are linked to (multiple) requirements. E.g. The ease of use (quality) of a method increases when automatic threat generation (requirement) is present.

Our study has made an hierarchical structure of qualities based on the definitions. An example of this structure can be found in Figure 1. In the upcoming questions, you will be asked to rank related concepts in order of their importance. To change the order of the concepts, simply **drag them up or down** until you are happy with the arrangement. If you come across any quality criteria that is not important at all, you can put them in last place and mention this at the bottom of the page.

Structure Figure 1: An example of the quality criteria structure.

Q1.1 As a decision-maker, could you rank these quality criteria groups in order of most important (1) to least important (n). (Hint: In the upcoming questions you can drag & drop each item)
______ Efficiency: The method can be performed at minimal cost and effort. (1)
______ Reliability: The method is semantically correct and meaningful. (2)
______ Applicability: The users are able to apply the method. (3)

Q1.2 Could you rank these quality criteria in the **efficiency** group in the order of most important (1) to least important (n).

______ Cost: The amount of resources to produce a threat model. Both the cost for creating a threat model (time/effort) and the cost for the purchase of a threat modeling tool are considered. (1)

______ Reusability: The ability to use parts of one threat model to create a new model. (2)

______ Tailorability: Capability of a method to adapt to a specific appliance. (3)

______ Scalability: If a method is applicable to large systems or not. There are components in the method that scale with the size and complexity of the system under assessment. (4)

Q1.3 Could you rank these quality criteria in the **reliability** group in the order of most important (1) to least important (n).

______ Completeness: Does the method specify a state or condition of having the necessary or appropriate parts of a threat model. (1)

______ Precision (ambiguity): How consistent the results of threat modeling are when conditions are the same. (2)

______ Documentation: The documentation of the method is clear and extensive. (3)

______ Software evolution support: How well a threat modeling method supports the continuous change and improvement of software throughout its lifecycle. (4)

______ Suitability: The aim of the method is inline with the expectations of the stakeholders. (5)

______ Maturity: How well the different parts of the method are structured and designed. (6)

______ Support for maintainability: How easy it is to make and track changes while using the method. (7)

______ Understandability: How easy is it is to understand the outcome of the method. (8)

Q1.3.1 Could you rank these quality sub-criteria related to **documentation** in the order of most important (1) to least important (n).

______ Technique documentation: The technique is documented clearly and extensively. (1)

______ Tool documentation: The tool has a clear and extensive documentation. (2)

Q1.3.2 Could you rank these quality sub-criteria related to **software evolution support** in the order of most important (1) to least important (n).

______ Modularity: How easy is it to develop and use software components separately. (1)
______ Component architecture: The level of support for a component-based structure that allows software modules to be added and removed with ease. (2)
______ Change propagation: Ability to keep track of changes made to system and to guarantee that a change is correctly propagated such that no inconsistent dependency is left unresolved. (3)
______ Change impact analysis: Ability to evaluate the effect that changes made to specific artefacts will have on other system components. (4)

Q1.3.3 Could you rank these quality sub-criteria related to **maturity** in the order of most important (1) to least important (n).

______ Technique maturity: The technique supported by a systematic and structured process. (1)
______ Tool maturity: Degree of optimization of the tool for standard usage. (Values based on the capability maturity model) (2)

Q1.3.4 Could you rank these quality sub-criteria related to **support for maintainability** in the order of most important (1) to least important (n).

______ Modifiable: The ability to make changes to an existing threat model. (1)
______ Traceability: Concerns the ability to trace the history, application and location of threat modeling components. (2)

Q1.4 Could you rank these quality criteria in the **applicability** group in the order of most important (1) to least important (n).

______ User experience: The experience of a person who uses a threat modeling method. (1)
______ Compatibility with agile development process: The technique can be used within an agile environment. (2)
______ Compatbility with related processes: How well the threat modeling method works with other (security) related processes such as risk assessment. (3)

Q1.4.1 Could you rank these quality sub-criteria related to **user experience** in the order of most important (1) to least important (n).
______ Ease of use: Simplicity of use of the technique, tool and notation. (1)
______ Learning curve: The cost associated with learning the threat modeling technique, tool and notation. (2)

Q1.5 Is there any quality (sub-)criteria you would like to add? If so, please explain briefly what it entails and where it should be placed.

____

____

____

____

____

Q1.6 Did you come across any quality (sub-)criteria that were not important or useful? (Please provide a short reasoning)

____

____

____

____

____

Q1.7 Do you think that any of the mentioned quality (sub-)criteria fits better in another group?

____

____

____

____

____

Q1.7 Do you have remaining questions, comments or uncertainties, you would like to share?

____

____

____

____

____

End of Block: Part 1: Quality criteria

Start of Block: Part 2: Requirements

Epic 1/8: Input. "Everything that happens before starting the threat analysis. This regards components that are related to identifying and describing what is already in place in the organization."

In this part you are asked to evaluate the requirements from the point of view of a **decision-maker**.

A requirement is an aspect of a threat modeling technique, tool or notation that a stakeholder might need.

The following definitions are to help you understand the difference between these concepts.
A **technique** is a stepwise procedure, possibly with a prescribed notation, to perform a development activity.
For example, a simple technique could include three steps: drawing system architecture, identifying threats and mitigating threats.
A **tool** is a possibly automated means to support a part of the development process.
For instance, Microsoft TMT is a tool that can be used to draw a system architecture and identify threats.
A **notation** is a system of symbols with a corresponding set of rules to construct artifacts used in communication.
An example of a notation is UML.
A **method** is the overarching concept containing a technique, tool and notation.

These concepts are often mentioned as a part of the requirement. Feel free to return to these definitions any time.

Each requirement is represented in a user story and is related to an epic. An epic is a group of user stories.

During the testrun of this survey we identified some complex terms. These are explained underneath the question. If there is a word you don't understand, please leave a comment at the end of the page. **If you do not understand the requirement,** please fill in **"no opinion"**.

This part has two types of questions:
1) As a decision-maker, evaluate if the criteria would be relevant for selecting a threat modeling method.
2) Based on your experience, assess which of the particular requirements are useful in real-world situations.

Q2.1 **Epic 1: Input. "Everything that happens before starting the threat analysis. This regards components that are related to identifying and describing what is already in place in the organization."**

Q2.1.1 I want the method to take current security measures into account, so irrelevant threats can be disregarded.
(E.g. DDOS related threats are disregarded due to DDOS protection implemented in organization)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.1.2a Input type: What type of input is required in order to start the threat analysis. (See the next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.1.2b Which of the following requirements related to **input type** is practically applicable in your experience? (multiple answers possible)

□ I want the analysis to require an attacker behavior* specification, so the analysis is based on what the system should be protected from. (1)

□ I want the analysis to require security assumptions*, so the analysis is based on assumptions about the system under analysis. (2)

□ I want the analysis to require system goals*, so that the analysis can be started as soon as a high-level description is provided. (3)

□ I want the analysis to require the architectural design of the system, so that the analysis is performed based on a model of the system. (4)

□ I want the analysis to be done based-on source code, so that the analysis is performed based on the actual system. (5)

□ None of them are practically applicable. (6)

□ Other: (7) ____

Attacker behavior: A list of expected actions and tactics used by a malicious actor to exploit vulnerabilities in computer systems.
Security assumptions: Underlying beliefs about the operational design of a system that can influence its security posture. In other words: Ideas about how a system should work that can affect how secure it is.
System goals: Intended outcomes or objectives that the system under analysis is designed to

achieve. These goals can be documented in a high-level description.

Q2.1.3a Input data: Where the input data comes from. (See the next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.1.3b Which of the following requirements related to **input data** is practically applicable in your experience? (multiple answers possible)

□ I want the threat assessment to be based-on expert opinion, so that no additional input data is required. (1)

□ I want the threat assessment to be based-on historical data. (2)

□ None of them are practically applicable. (3)

□ Other: (4) ____

Comments 2.1 Do you have remaining questions, comments or uncertainties, you would like to share?

____

____

____

____

____

Epic 2/8: Output. "Requirements related to the outcome of the threat modeling method."

Warning: The following paragraph is repetitive, so you only need to read it again if you have forgotten any of the details. You can skip to the questions.

In this part you are asked to evaluate the requirements from the point of view of a **decision-maker**.

A requirement is an aspect of a threat modeling technique, tool or notation that a stakeholder might need.

The following definitions are to help you understand the difference between these concepts.
A **technique** is a stepwise procedure, possibly with a prescribed notation, to perform a development activity.
For example, a simple technique could include three steps: drawing system architecture, identifying threats and mitigating threats.
A **tool** is a possibly automated means to support a part of the development process.
For instance, Microsoft TMT is a tool that can be used to draw a system architecture and identify threats.
A **notation** is a system of symbols with a corresponding set of rules to construct artifacts used in communication.
An example of a notation is UML.
A **method** is the overarching concept containing a technique, tool and notation.

These concepts are often mentioned as a part of the requirement. Feel free to return to these definitions any time.

Each requirement is represented in a user story and is related to an epic. An epic is a group of user stories.

During the testrun of this survey we identified some complex terms. These are explained underneath the question. If there is a word you don't understand, please leave a comment at the end of the page. **If you do not understand the requirement**, please fill in **"no opinion"**.

This part has two types of questions:
1) As a decision-maker, evaluate if the criteria would be relevant for selecting a threat modeling method.
2) Based on your experience, assess which of the particular requirements are useful in real-world situations.

Q2.2 **Epic 2: Output. "Requirements related to the outcome of the threat modeling method."**

Q2.2.1a Output type: The type of artifact that the method produces. (See the next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.2.2b Which of the following requirements related to **output type** is practically applicable in your experience? (multiple answers possible)

□ I want the threat modeling method to produce security threats. (1)

□ I want the threat modeling method to produce threat mitigations, so general risk-lowering countermeasures are obtained. (2)

□ I want the threat modeling method to produce security requirements, so that a refined set of specific countermeasures is obtained. (3)

□ None of them are practically applicable. (4)

□ Other: (5) ____

2.2.3a Output granularity: The level of detail in which the result from threat modeling is presented. (See the next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.2.3b Which of the following requirements related to **output granularity** is practically applicable in your experience? (multiple answers possible)

□ I want the threat modeling method to produce a high-level outcome, so a general overview can be obtained that can guide the action points. (An example of a high level outcome is a document that provides an overview of general security improvements.) (1)

□ I want the threat modeling method to produce a low-level outcome, so that the outcome can be directly used for implementation. (An example of a low-level outcom is a model of the improved system, which can directly be translated into code.) (2)

□ None of them are practically applicable. (3)

□ Other: (4) ____

Q2.2.4b Report representation: How the findings from threat modeling are displayed in the report. (See the next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.2.4b Which of the following requirements related to **report representation** is practically applicable in your experience? (multiple answers possible)

□ I want the report from a threat modeling project to-be a structured text. (An example of structured text can be a .csv file). (1)

□ I want the report from a threat modeling project to have a model-based representation. (2)

□ None of them are practically applicable. (3)

□ Other: (4) ____

Q2.2.5a I want the method to assist in providing mitigation strategies.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Comments 2.2 Do you have remaining questions, comments or uncertainties, you would like to share?

____

____

____

____

____

Page Break

Epic 3/8: Perspective. "Requirements related to the perspective of the threat modeling method. Contains components that represent the point of view from which the approach solves the threat modeling case"

Warning: The following paragraph is repetitive, so you only need to read it again if you have forgotten any of the details. You can skip to the questions.

In this part you are asked to evaluate the requirements from the point of view of a **decision-maker**.

A requirement is an aspect of a threat modeling technique, tool or notation that a stakeholder might need.

The following definitions are to help you understand the difference between these concepts.
A **technique** is a stepwise procedure, possibly with a prescribed notation, to perform a development activity.
For example, a simple technique could include three steps: drawing system architecture, identifying threats and mitigating threats.
A **tool** is a possibly automated means to support a part of the development process.
For instance, Microsoft TMT is a tool that can be used to draw a system architecture and identify threats.
A **notation** is a system of symbols with a corresponding set of rules to construct artifacts used in communication.
An example of a notation is UML.
A **method** is the overarching concept containing a technique, tool and notation.

These concepts are often mentioned as a part of the requirement. Feel free to return to these definitions any time.

Each requirement is represented in a user story and is related to an epic. An epic is a group of user stories.

During the testrun of this survey we identified some complex terms. These are explained underneath the question. If there is a word you don't understand, please leave a comment at the end of the page. **If you do not understand the requirement**, please fill in **"no opinion"**.

This part has two types of questions:
1) As a decision-maker, evaluate if the criteria would be relevant for selecting a threat modeling method.
2) Based on your experience, assess which of the particular requirements are useful in real-world situations.

Q2.3 **Epic 3: Perspective. "Requirements related to the perspective of the threat modeling method. Contains components that represent the point of view from which the approach solves the threat modeling case"**

Q2.3.1a Approach: The central ideology that the threat modeling method is based on. (See next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.3.1b Which of the following requirements related to **approach** is practically applicable in your experience? (multiple answers possible)

□ I want the technique to be attack-centric, so there will be a focus on identifying attacker profiles and the complexity of attacks. (1)

□ I want the technique to be risk-centric, so that impact and likelihood of the threats decide which security requirement needs to be addressed first. (2)

□ I want the technique to be software-centric, so the focus will be on the software that is examined. (E.g. STRIDE analysis on DFDs) (3)

□ I want the technique to be centered around Goal-Oriented Requirement Engineering(GORE), so that both functional and non-functional (among other, security) requirements are obtained. (4)

□ I want the technique to be centered around Security Requirements Engineering (SRE), so that the primary goal is to identify security requirements. (5)

□ None of them are practically applicable. (6)

□ Other: (7) ____

Q2.3.2a Focus: The viewpoint from which the technique is performed. (See next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.3.2b Which of the following requirements related to **focus** is practically applicable in your experience? (multiple answers possible)

□ I want the technique to be designed from the point of view of the attacker. (1)

□ I want the technique to be designed from the point of view of a defender. (2)

□ None of them are practically applicable. (3)

□ Other: (4) ____

Q2.3.3a Agents: Does a threat model focus on one perspective or multiple perspectives at the same time. (only relevant for attack-centric threat models).

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.3.3b Which of the following requirements related to **agents** is practically applicable in your experience? (multiple answers possible)

□ I want one model to show the perspectives of multiple agents, so that conflicting perspectives can be considered. (For attack-centric approaches) (1)

□ I want one model to show the perspective of a single agent, so that the model does not become too complicated. (For attack-centric approaches) (2)

□ None of them are practically applicable. (3)

□ Other: (4) ____

Q2.3.4a Modeling view: The level of detail on which the actual model is created. (See next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.3.4b Which of the following requirements related to **modeling view** is practically applicable in your experience? (multiple answers possible)

□ I want the threat model to describe the architecture of the system, so that threats can be related to different parts of the system. (1)

□ I want the threat model to describe the functional behavior of the system*. (2)

□ I want the threat model to describe the platform of the system, so that operating systems, middleware and also physical parts are considered. (3)

□ None of them are practically applicable. (4)

□ Other: (5) ____

Functional behavior of the system refers to the intended or expected actions and outputs of the system. For example, a payment system transfers a balance from one account to the other, using the internet. When there is a DDOS attack on the system, this threat results in the function being denied. Meaning that the system is not able to transfer balance.

Comments 2.3 Do you have remaining questions, comments or uncertainties, you would like to share?

____

____

____

Page Break

Intro Epic 4/8 Notation: "Requirements specific to the notation used to represent the threat model."

Warning: The following paragraph is repetitive, so you only need to read it again if you have forgotten any of the details. You can skip to the questions.

In this part you are asked to evaluate the requirements from the point of view of a **decision-maker**.

A requirement is an aspect of a threat modeling technique, tool or notation that a stakeholder might need.

The following definitions are to help you understand the difference between these concepts.
A **technique** is a stepwise procedure, possibly with a prescribed notation, to perform a development activity.
For example, a simple technique could include three steps: drawing system architecture, identifying threats and mitigating threats.
A **tool** is a possibly automated means to support a part of the development process.
For instance, Microsoft TMT is a tool that can be used to draw a system architecture and identify threats.
A **notation** is a system of symbols with a corresponding set of rules to construct artifacts used in communication.
An example of a notation is UML.
A **method** is the overarching concept containing a technique, tool and notation.

These concepts are often mentioned as a part of the requirement. Feel free to return to these definitions any time.

Each requirement is represented in a user story and is related to an epic. An epic is a group of user stories.

During the testrun of this survey we identified some complex terms. These are explained underneath the question. If there is a word you don't understand, please leave a comment at the end of the page. **If you do not understand the requirement**, please fill in **"no opinion"**.

This part has two types of questions:
1) As a decision-maker, evaluate if the criteria would be relevant for selecting a threat modeling method.
2) Based on your experience, assess which of the particular requirements are useful in real-world situations.

Q2.4 **Epic 4 Notation: "Requirements specific to the notation used to represent the threat model."**

Q2.4.1a Notation abstraction: The level of detail at which the components are represented in the notation. (See next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.4.1b Which of the following requirements related to **notation abstraction** is practically applicable in your experience? (multiple answers possible)

□ I want the notation to represent its components surrounding (security) goals*. (1)

□ I want the notation to represent its components surrounding a model of the system. (2)

□ I want the notation to represent its components surrounding (security) problems*. (3)

□ I want the notation to represent its components surrounding the business processes. (4)

□ None of them are practically applicable. (5)

□ Other: (6) ____

(Security) Goals: Intended (security) outcomes or objectives that the system under analysis is designed to achieve. An example of a security goal can be: To ensure that only authorized users are able to access the information (confidentiality).
(Security) Problem: Vulnerability in a system that can be exploited by attackers. For example,

using TLS 1.0 in your system is a security problem.

Q2.4.2a Formality: Distinguishes between formal (textual) and graphical representations. (See next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.4.2b Which of the following requirements related to **formality** is practically applicable in your experience? (multiple answers possible)

□ I want the notation to be formal, so that threat modeling based on a mathematical model is possible. (1)

□ I want the notation to be graphical, so that attack trees, attack graphs, Data Flow Diagrams (DFDs), or tables can be used. (2)

□ None of them are practically applicable. (3)

□ Other: (4) ____

Q2.4.3 I want to use a notation for threat modeling that I am familiar with. (E.g. DFD, petri-nets, UML)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.4.4 I want to use (an extension of) a notation that is a well-known software specification language, so that threat modeling becomes easier to learn.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.4.5 I want to be able to extend the modeling notation, so that domain-specific components can be modeled.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.4.6 Could you list the components that must be present in a threat modeling notation? (For example, threats, mitigations, etc.)

____

Q2.4.6 I want the notation to have a specific set of components present. (Based on previous question)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

Q2.4.7 I want to have a notation where the most common relationships between components are present.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.4.8 I want to be able to visualize how the proposed countermeasures affect the system's weaknesses, so I can understand how effective the mitigations are.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Comments 2.4 Do you have remaining questions, comments or uncertainties, you would like to share?

____

____

____

____

____

Page Break

Epic 5/8: Modeling context. "These requirements are about the context in which the modeling method can or cannot be applied, which is called the scope of usage."

Warning: The following paragraph is repetitive, so you only need to read it again if you have forgotten any of the details. You can skip to the questions.

In this part you are asked to evaluate the requirements from the point of view of a **decision-maker**.

A requirement is an aspect of a threat modeling technique, tool or notation that a stakeholder might need.

The following definitions are to help you understand the difference between these concepts.
A **technique** is a stepwise procedure, possibly with a prescribed notation, to perform a development activity.
For example, a simple technique could include three steps: drawing system architecture, identifying threats and mitigating threats.
A **tool** is a possibly automated means to support a part of the development process.
For instance, Microsoft TMT is a tool that can be used to draw a system architecture and identify threats.
A **notation** is a system of symbols with a corresponding set of rules to construct artifacts used in communication.
An example of a notation is UML.
A **method** is the overarching concept containing a technique, tool and notation.

These concepts are often mentioned as a part of the requirement. Feel free to return to these definitions any time.

Each requirement is represented in a user story and is related to an epic. An epic is a group of user stories.

During the testrun of this survey we identified some complex terms. These are explained underneath the question. If there is a word you don't understand, please leave a comment at the end of the page. **If you do not understand the requirement**, please fill in **"no opinion"**.

This part has two types of questions:
1) As a decision-maker, evaluate if the criteria would be relevant for selecting a threat modeling method.
2) Based on your experience, assess which of the particular requirements are useful in real-world situations.

Q2.5 **Epic 5: Modeling context. "These requirements are about the context in which the modeling method can or cannot be applied, which is called the scope of usage."**

Q2.5.1 I want a threat modeling method that does not have any **knowledge barriers***.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Hard terms Q2 **Knowledge barriers** are obstacles or limitations that prevent practitioners from being able to perform threat modeling. For example, a cursus is required to perform a specific threat modeling method.

Q2.5.2a Layer: The level of system details at which threats are assessed. (See next question, for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.5.2b Which of the following requirements related to **layer** is practically applicable in your experience? (multiple answers possible)

□ I want the threat modeling method to identify threats on the software level, so that only software components are considered. (1)

□ I want the threat modeling method to identify threats on the computer-based system level, so that both software and hardware are considered. (2)

□ I want the threat modeling method to identify threats on the total system level, so that man, technology and organization (MTO) are considered. (3)

□ I want the threat modeling method to identify threats on the environment level, so that factors beyond MTO are considered. (4)

□ None of them are practically applicable. (5)

□ Other: (6) ____

Q2.5.3a Applicability to the system: The method is designed for either general use or a specific type of computer-based system. (See next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.5.3b Which of the following requirements related to **applicability to system** is practically applicable in your experience? (multiple answers possible)

□ I want to have a method that is generally applicable, so I can use it in many different systems. (1)

□ I want to have a method that is specifically designed for the system under analysis. (An example of a specific method would be: a threat modeling method for cloud applications) (2)

□ None of them are practically applicable. (3)

□ Other: (4) ____

Q2.5.4 I want to have a threat modeling method that considers threats to hardware. (E.g. physical tampering is considered)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Comments 2.5 Do you have remaining questions, comments or uncertainties, you would like to share?

____

____

____

____

____

Page Break

Epic 6/8: Method process. "Requirements that apply to the process of threat modeling and its environment.

Warning: The following paragraph is repetitive, so you only need to read it again if you have forgotten any of the details. You can skip to the questions.

In this part you are asked to evaluate the requirements from the point of view of a **decision-maker**.

A requirement is an aspect of a threat modeling technique, tool or notation that a stakeholder might need.

The following definitions are to help you understand the difference between these concepts.
A **technique** is a stepwise procedure, possibly with a prescribed notation, to perform a development activity.
For example, a simple technique could include three steps: drawing system architecture, identifying threats and mitigating threats.
A **tool** is a possibly automated means to support a part of the development process.
For instance, Microsoft TMT is a tool that can be used to draw a system architecture and identify threats.
A **notation** is a system of symbols with a corresponding set of rules to construct artifacts used in communication.
An example of a notation is UML.
A **method** is the overarching concept containing a technique, tool and notation.

These concepts are often mentioned as a part of the requirement. Feel free to return to these definitions any time.

Each requirement is represented in a user story and is related to an epic. An epic is a group of user stories.

During the testrun of this survey we identified some complex terms. These are explained underneath the question. If there is a word you don't understand, please leave a comment at the end of the page. **If you do not understand the requirement**, please fill in **"no opinion"**.

This part has two types of questions:
1) As a decision-maker, evaluate if the criteria would be relevant for selecting a threat modeling method.
2) Based on your experience, assess which of the particular requirements are useful in real-world situations.

Q2.6 **Epic 6: Method process. "Requirements that apply to the process of threat modeling and its environment.**

Q2.6.1a Threat generation: The process of identifying and establishing a list of potential threats.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.6.1b Which of the following requirements related to **threat generation** is practically applicable in your experience? (multiple answers possible)

□ I want to have a method that generates threats based on a knowledge base/methodology. (e.g. Threat generation using STRIDE) (1)

□ I want to have a method that generates threats based on expert knowledge. (e.g. CORAS) (2)

□ None of them are practically applicable. (3)

□ Other: (4) ____

Q2.6.2 I want to have a threat modeling method that supports the participation of specific stakeholders. (e.g. Threat modeling by security specialist versus threat modeling by developer)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.6.3 I want the technique to clearly explain how to define the limits of the system, so that we can identify the area that needs protection. (E.g. only consider assets with a specific priority level)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Comments 2.6 Do you have remaining questions, comments or uncertainties, you would like to share?

____

____

____

____

____

Page Break

Intro Epic 7/8: Analysis. "Requirements that are related to the threat analysis. This mostly regards the steps involved, the reasoning used, and the aspects to be considered during the analysis."

Warning: The following paragraph is repetitive, so you only need to read it again if you have forgotten any of the details. You can skip to the questions.

In this part you are asked to evaluate the requirements from the point of view of a **decision-maker**.

A requirement is an aspect of a threat modeling technique, tool or notation that a stakeholder might need.

The following definitions are to help you understand the difference between these concepts.
A **technique** is a stepwise procedure, possibly with a prescribed notation, to perform a development activity.
For example, a simple technique could include three steps: drawing system architecture, identifying threats and mitigating threats.
A **tool** is a possibly automated means to support a part of the development process.
For instance, Microsoft TMT is a tool that can be used to draw a system architecture and identify threats.
A **notation** is a system of symbols with a corresponding set of rules to construct artifacts used in communication.
An example of a notation is UML.
A **method** is the overarching concept containing a technique, tool and notation.

These concepts are often mentioned as a part of the requirement. Feel free to return to these definitions any time.

Each requirement is represented in a user story and is related to an epic. An epic is a group of user stories.

During the testrun of this survey we identified some complex terms. These are explained underneath the question. If there is a word you don't understand, please leave a comment at the end of the page. **If you do not understand the requirement**, please fill in **"no opinion".**

This part has two types of questions:
1) As a decision-maker, evaluate if the criteria would be relevant for selecting a threat modeling method.
2) Based on your experience, assess which of the particular requirements are useful in real-world situations.

Q2.7 **Epic 7: Analysis. "Requirements that are related to the threat analysis. This mostly regards the steps involved, the reasoning used, and the aspects to be considered during the analysis."**

Q2.7.1 I want a threat modeling method that differentiates between critical and non-critical components.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.7.2 I want the threat modeling method to show how attacks on one part of the system can affect other parts that depend on it. (e.g. internet facing login application is compromised --> show effect on user database)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.7.3 I want a technique that ensures the quality of the outcome as part of the analysis procedure. (e.g. Use guidelines for quality assessment of the model)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.7.4 I want a method that uses a list of known threats (a threat library) to identify applicable security threats based on the system's components.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.7.5a Analysis level: The level of detail used to analyze the system for potential threats. (See next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.7.5b Which of the following requirements related to **analysis level** is practically applicable in your experience? (multiple answers possible)

□ I want the analysis to be done at the source-code level. (1)

□ I want the analysis to be done at system design level. (In addition to the system's structure, the system design level also covers security controls' technical aspects, like encryption standards) (2)

□ I want the depth of the analysis to reach architecture level, so it can be executed early on in the development process. (3)

□ None of them are practically applicable. (4)

□ Other: (5) ____

Q2.7.6a Security objective: The areas of security covered by the method. (See next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.7.6b Which of the following requirements related to **security objective** is practically applicable in your experience? (multiple answers possible)

□ I want the security objective of the threat modeling method to cover confidentiality. (In other words, the threat modeling method can analyse threats to confidentiality) (1)

□ I want the security objective of the threat modeling method to cover integrity. (2)

□ I want the security objective of the threat modeling method to cover availability. (3)

□ None of them are practically applicable. (4)

□ Other: (5) ____

Q2.7.7 I want the technique to prioritize threats, so that we can determine the order in which to implement the countermeasures.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.7.8a Risk: The role of risk in the threat modeling technique. Risk is expressed in terms of likelihood and impact of a malicious event. (See next question for examples)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.7.8b Which of the following requirements related to **risk** is practically applicable in your experience? (multiple answers possible)

□ I want to have a technique that connects the level of risk to the identified threats, so that threats can be prioritized. (1)

□ I want to have a technique that considers risk externally. (E.g. by combining the technique with an external risk management framework) (2)

□ I want to have a technique that does not consider risk. (E.g. because it is not needed to obtain the prefered threat modeling outcome) (3)

□ None of them are practically applicable. (4)

□ Other: (5) ____

Q2.7.9 I want to have a threat modeling method that considers **steps to vulnerability exploitation*** as part of the analysis.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Steps to vulnerability exploitation refers to the steps an attacker may take to identify and exploit a vulnerability. For this research we measure this in term of estimated time until a vulnerability is exploited.

Q2.7.10a Threat analysis type: The differentiation between quantitative and qualitative techniques for threat analysis. (The definitions are given udnerneath the next question)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.7.10b Which of the following requirements related to **threat analysis type** is practically applicable in your experience? (multiple answers possible)

□ I want my threat modeling analysis to be qualitative, so it is faster. (1)

□ I want my threat modeling method to be quantitative, so it is more detailed. (2)

□ None of them are practically applicable. (3)

□ Other: (4) ____

Qualitative threat modeling methods are more subjective, often based on brainstorming, expert opinion and non quantifiable techniques. It does not require much data and details but rather relies on diagrams and checklists.

Quantitative threat modeling is based on data and statistical analysis in order to quantify the effect of security threats. This often involves mathematical models or simulations of different types of attacks.

Comments 2.7 Do you have remaining questions, comments or uncertainties, you would like to share?

____

____

____

____

____

Page Break

Epic 8/8: Other. "Requirements that are not placed in a category, explicitly relate to the integration with the SDLC, or relate to tool support."

Warning: The following paragraph is repetitive, so you only need to read it again if you have forgotten any of the details. You can skip to the questions.

In this part you are asked to evaluate the requirements from the point of view of a **decision-maker**.

A requirement is an aspect of a threat modeling technique, tool or notation that a stakeholder might need.

The following definitions are to help you understand the difference between these concepts.
A **technique** is a stepwise procedure, possibly with a prescribed notation, to perform a development activity.
For example, a simple technique could include three steps: drawing system architecture, identifying threats and mitigating threats.
A **tool** is a possibly automated means to support a part of the development process.
For instance, Microsoft TMT is a tool that can be used to draw a system architecture and identify threats.
A **notation** is a system of symbols with a corresponding set of rules to construct artifacts used in communication.
An example of a notation is UML.
A **method** is the overarching concept containing a technique, tool and notation.

These concepts are often mentioned as a part of the requirement. Feel free to return to these definitions any time.

Each requirement is represented in a user story and is related to an epic. An epic is a group of user stories.

During the testrun of this survey we identified some complex terms. These are explained underneath the question. If there is a word you don't understand, please leave a comment at the end of the page. **If you do not understand the requirement**, please fill in **"no opinion"**.

This part has two types of questions:
1) As a decision-maker, evaluate if the criteria would be relevant for selecting a threat modeling method.
2) Based on your experience, assess which of the particular requirements are useful in real-world situations.

Q2.8 **Epic 8: Other. "Requirements that are not placed in a category, explicitly relate to the integration with the SDLC, or relate to tool support."**

Q2.8.1 I want to have a definition of done (stopping condition) in the technique, so users are uniformly guided on when the threat modeling is complete.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.8.2 I want a threat modeling method that can work together with another cyber security method. (E.g. Can the results of threat modeling be used in another security method?, or is the threat modeling method a part of a group of security methods?)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.8.3 I want to receive validation support for the threat model, so that I can be more sure that it accurately reflects the real-world situation. (e.g. The tool provides recommendations and feedback about common mistakes during the model development).

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.8.4 I want to receive verification support for the threat model, so I can guarantee the model is built correctly. (e.g. Only allow relationships between components when they are meaningful)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.8.5 I want to have a tool that assists the threat modeling technique by (semi-)automating some of the steps.

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Q2.8.6 I want a tool that can transfer projects from one device to another. (For example, through an export and import function.)

○ Relevant when deciding between threat modeling methods (1)

○ Not relevant (2)

○ No opinion (3)

Comments 2.8 Do you have remaining questions, comments or uncertainties, you would like to share?

____

____

____

End of Block: Part 2: Requirements